AS
ASCII Learning System

アスキー・ラーニングシステム　①入門コース

入門 MS-DOS

村瀬康治 著

改訂新版

アスキー出版局

アスキー・ラーニングシステム
入門 MS-DOS
改訂新版
村瀬康治 著
①入門コース

ASCII
LEARNING SYSTEM

アスキー・ラーニングシステム　①入門コース

入門 MS-DOS

村瀬康治 著

改訂新版

アスキー出版局

MS-DOS Learning System 全3巻の構成

アスキー・ラーニングシステムMS-DOS編は、『入門MS-DOS』『実用MS-DOS』『応用MS-DOS』の3部から構成されています。

入門編および実用編では、ビジネスソフトのユーザーをはじめとする、広く一般的なユーザーを対象に、MS-DOSの基礎知識とその活用法を初歩からわかりやすく解説します。また応用編では、MS-DOS上で各種のソフトウェアを開発するプログラマー、あるいはMS-DOSの内部を詳しく知りたいユーザーを対象に、MS-DOSシステムの初歩から、アセンブラやC言語による実際のプログラミング例を豊富にまじえながら、高度な知識に至るまでを解説します。

それぞれの巻は次のように構成されています。

入門MS-DOS：MS-DOS上の各種のビジネスソフトなどを利用しようとする人を対象に、MS-DOSの基本的な機能や操作法をやさしく、ていねいに解説します。とくに、MS-DOSによって初めてOSに接する人も、楽に読み進めるよう十分配慮されています。この入門編により、各種のソフトを活用するためのMS-DOSの知識が、バランスよく得られるようにまとめてあります。

実用MS-DOS：MS-DOSの基礎知識を学んだ人に対して、MS-DOSマシン本来の実力をフルに発揮させ、各種のソフトを効果的に使いこなすための知識とノウハウを提供します。ATOK、松茸、VJEなどによる日本語入力システムの使い方や漢字コードの知識、ハードディスクの各形態のフォーマット処理や初期設定、ハードディスクの効果的運用法、RAMディスクやディスクキャッシュ、それにEMSを実現する具体例とその利用法、さらに各種エディタのやさしい使い方などを実例解説します。また、目的別によるMS-DOSの主要コマンドの使い方を、コマンドリファレンスの形式と実行例により示します。この実用編は、ビジネスソフトのユーザーにも、ソフトウェア開発者にも、MS-DOSマシンを使いこなす重要な知識を提供するでしょう。

応用MS-DOS：MS-DOS上でソフトウェアを開発する人や、MS-DOSの内部構造を詳しく知りたい人に対して、豊富な図や実際のプログラムを示しながら、MS-DOSシステムの構造をその基本から細部まで、ていねいに解説します。また、システムコールの解説や、アセンブラによるソフトウェア開発、さらにC言語の開発環境について解説し、そのプログラミングを実習します。この応用編は、これからMS-DOSシステムを学ぶプログラマーにとって、たいへん親切に書かれた最適の参考書となるでしょう。

このシリーズは、MS-DOSのバージョン3.3をもとに書かれていますが、それ以前のバージョンをお使いの方も、まったく支障なく利用できます。このMS-DOS 3部作を読むにあたり、MS-DOSマシンを操作できれば理想的ですが、いずれも豊富な実行例をもとに解説していますので、紙上体験だけでも大きな効果が得られるでしょう。

はじめに

この『入門 MS-DOS』を含む MS-DOS 3 部作は、初版より 6 年目を迎え、その発行部数(3 部合計)は 50 万部を超えました。こんなにも多くのみなさんに愛読され、たいへんうれしく思う反面、一層大きな「責任」を感じています。そこで、これを機会に、時代の進歩やユーザーのニーズに合わせて、3 部作を全面改訂いたしました。50 万部の責任と信頼に応えるべく、より実用性が高く、みなさんに喜んでいただける参考書に生まれ変わったと自負しています。

MS-DOS が、IBM のパーソナル・コンピュータ用の OS「PC-DOS」として誕生したのは、ほぼ一昔も前の 1981 年のことですが、最近になって、MS-DOS の寿命が今後さらに延びるという予測が、いろいろな事情からはっきりしてきました。MS-DOS は、16 ビットマシンのための OS ですが、次世代のマシンである 32 ビットマシンと、そこで使われる MS-OS/2 という MS-DOS の後継 OS が登場した後も、まだまだ当分の間は〝今まで以上に″いそがしく活躍し続けることでしょう。このことは、8 ビットマシンの標準的 OS であった CP／M の時代が、16 ビットマシンの MS-DOS の時代に、短期間に移行したのとは対象的です。この違いは、MS-DOS マシンが、今のままでも、仕事によってはまだまだ十分に使える実力を持った、バランスのよいシステムであることによります。結局 MS-DOS は、次世代のマシンが普及した後も、ずっと平行して使われ続ける可能性が非常に高いのです。

さて、本書を手に取られたみなさんには、パーソナル・コンピュータの「オペレータ」ではなく、ぜひ「活用者」になっていただきたいと思います。そのために、ぜひ MS-DOS を学んでください。一太郎や 1-2-3 の世界だけで仕事をする人と、MS-DOS の世界を知り、その力を仕事に利用できる人とでは、「仕事」の質が決定的に違ってくるでしょう。

MS-DOS を学ぶことによる利益は、すぐに表れます。MS-DOS に接することにより、コンピュータの本質が明らかになり、コンピュータの基本、ソフトウェアの基本が理解できるようになりますし、それによってコンピュータによる「情報処理」というものがわかってきます。つまり MS-DOS の知識が、これから学んでいくコンピュータのすべての知識の素になるのです。

いまMS-DOSを知ることは、これからのコンピュータの世界を歩いて行く上で、最初のキップを手にすることに等しく、実に大きな意味を持つことになるでしょう。しかし、ビジネスソフトを利用するユーザーの場合は、MS-DOSを隅々まで知る必要はありません。まずは、本書を片手に、自分流に楽しく学んでいってください。本書は、自分の仕事にパーソナル・コンピュータを利用しようと考えている普通の人が読むことを想定し、できる限りわかりやすく、かつ本質的な理解も得られるように、バランス良くていねいに書いたつもりです。

本書の知識があれば、パーソナル・コンピュータに対してどれほど自信が持てるようになるか、きっと多くの人に分っていただけるのではないかと思っています。そして、本書がMS-DOSを知るための中心的役割を、今まで以上に果してくれることを願っています。

1989年10月　村瀬康治

目　　次

4章 システムディスクとデータディスク

5章 コントロールキーによる便利な機能

6章 ファイル/ファイル名/ファイルの種類

7章 重要コマンドとその機能

11章　ファイルの組織的な分類整理法

12章　フィルタとパイプおよびリダイレクト

13章　MS-DOS 上の各種ソフトウェア

APPENDIX

イラスト：長谷川 弘

1章
パソコン・ユーザーと MS-DOSとのかかわり

ソフトウェアシーンは今、MS-DOS の時代

現在の最も一般的なパーソナル・コンピュータである、16 ビットマシン上で使われている実務ソフトのほとんどすべては、「MS-DOS」という「基本ソフトウェア」(Operating System：OS)上で動作するように作られています。本格的な実務ソフトの世界は、BASIC(プログラミングの入門用言語の１つ)の世界とは関係のない、MS-DOS で代表される OS を基盤とする世界です。

ソフトウェアの開発に携わる専門家は当然のことながら、一般のビジネスソフトの利用者にとっても、OS の知識は非常に重要です。つまり、コンピュータを自在に活用するためには、OS の基礎知識がどうしても必要になるのです。

そこでまず本章では、MS-DOS という OS が、何をする場合にも「基本ソフトウェア」として存在することを認識するとともに、OS がなぜ必要なのかを解説します。

MS-DOS ―― つまり OS とは何かということを、今すぐ理解する必要はありません。そう簡単に説明や理解ができる性質のものではないのです。しかし、本書を読み進んでいくうちに、「コンピュータのハードウェア」と、「実行するソフトウェア」、それに「私たちユーザー」とを有機的に結びつけ、コンピュータに生命を与える「オペレーティングシステム(OS)」という基本ソフトウェアの概念が、徐々に理解されてくるでしょう。

1.1 気付かないで使っているMS-DOS

16ビットパーソナル・コンピュータを実務に使っている私たち、その大半の人たちは、MS-DOSという「基本ソフトウェア」(**オペレーティングシステム**：各種のプログラムを実行するためのベースとなるソフトウェア)のお世話になっています。しかし、現実にはこのMS-DOSの存在に気付かないで、あるいは知っていても、とくにその意味を意識することなく仕事をしている人が多いのではないでしょうか。

ビジネスソフトを例にとると、

日本語ワードプロセッサ ……… 一太郎、新松、P1.EXE
表計算ソフト …………………… Multiplan、Lotus 1-2-3
データベースソフト …………… informix-SQL、dBASE IV

などで代表される実務向けソフトはもとより、ほとんどのビジネスソフトは、基本ソフトウェアであるMS-DOSがコンピュータ上で働いていてこそ、実行が可能となります。しかしこれらのソフトが起動し、それぞれの世界にはいってしまえば、MS-DOSは表面にほとんどその姿を見せません。

ここで、代表的なビジネスソフトの1つである表計算ソフトのMultiplanを実際に立ち上げてみましょう。Multiplanの日常使っているディスクをドライブA：(ドライブ1)にセットして、リセットボタンを押します(図 1.1)。

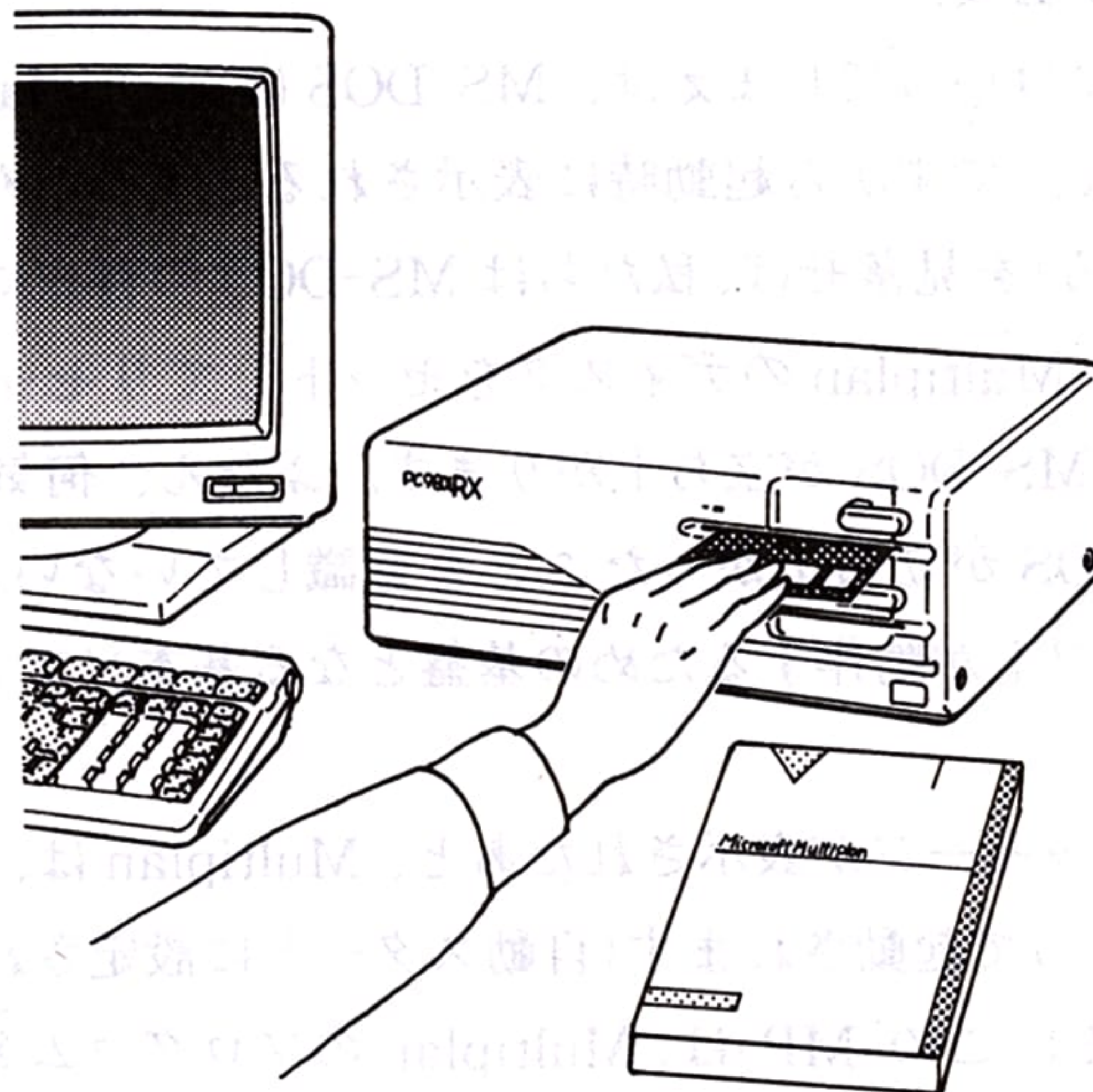



図 1.1　日常使っているMultiplanのディスクをセットする

リセットボタンを押すことにより(あるいはコンピュータの電源を ON することにより)、コンピュータのさまざまな動作が開始され、ディスクが作動中であることを示すランプが何回か点滅します。そのときの画面を注意深く観察してみましょう。Multiplan が立ち上がるまでの短い時間に、次のようなメッセージが表示されます(**図 1.2**)。

```
(ディスクをセットしてリセットボタンを押す。あるいは電源ON)
NEC PC-9800 Series Personal Computer

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30                          } MS-DOSのオープニングメッセージ
Copyright 1981,88 Microsoft Corp. / NEC Corporation

自由文変換システム　ＡＴＯＫ６ ver 1.2
Copyright 1986,87 (株)ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 3.30

A>MP              このあと、自動的にMultiplanの起動が開始される
```

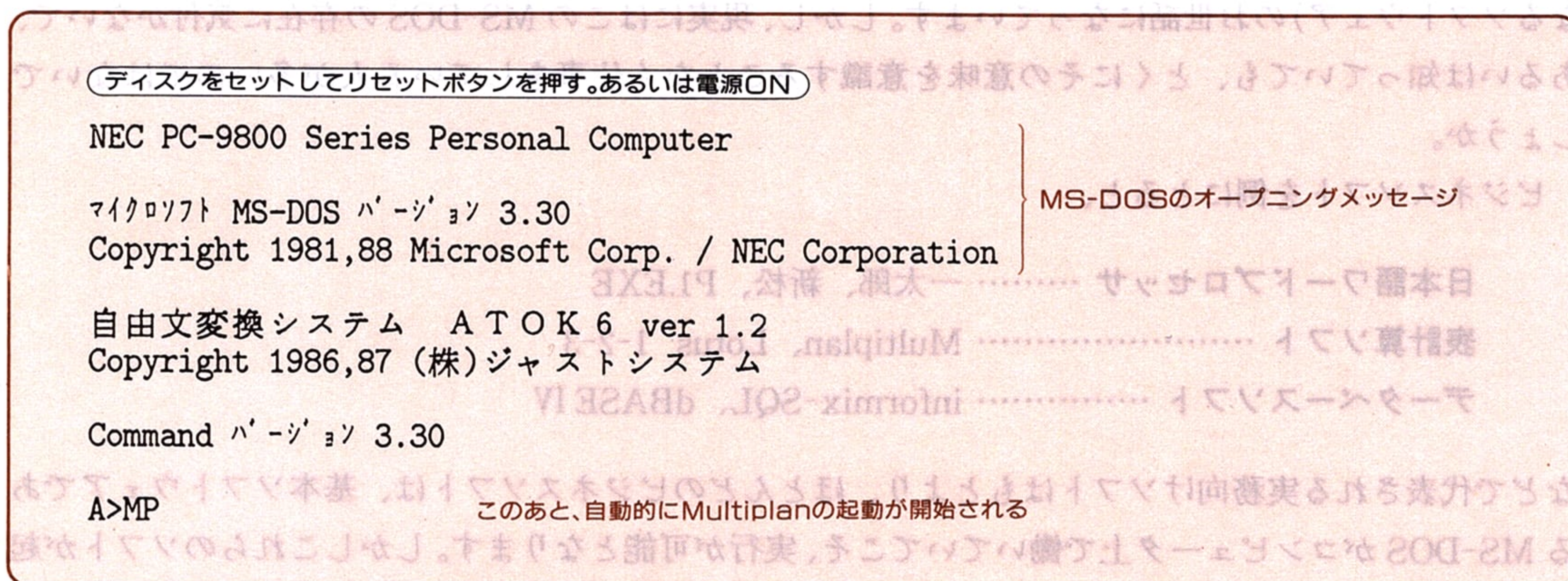



図 1.2　Multiplan のディスクから、まず最初に MS-DOS が起動する

この Multiplan のディスクは、電源を ON するか、リセットボタンを押すだけで、あとは自動的に Multiplan が立ち上がるようになっており、この表示のあと、直ちにその起動がかかり、Multiplan の世界にはいります。このような起動のしかたを**自動スタート**と呼びます(自動スタートになっていない場合は、このあと若干の操作が必要)。

いったん Multiplan の世界にはいってしまえば、MS-DOS は Multiplan を「終了」するまでほとんど顔を見せることはありません。ですから起動時に表示される図 1.2 のメッセージ(MS-DOS の**オープニングメッセージ**と呼ばれる)を見落せば、私たちは MS-DOS の存在に気付かないかも知れません。

図 1.2 からもわかるように、Multiplan のディスクをセットしてリセットボタンを押すと、Multiplan が立ち上がる前に、まず MS-DOS が立ち上がります。ふだん、何気なくこの作業を行っている私たちは、この時点で MS-DOS が立ち上がったことを意識していないかも知れません。しかし、Multiplan をはじめ、各種のソフトが動作するための基盤となる基本ソフトウェアが、ここで起動したのです。

MS-DOS のオープニングメッセージが表示されたあと、Multiplan は、自動スタートの機能によって入力される「A>MP ⏎」によって起動されます(自動スタートに設定されてない場合は、これをキーボードから入力する必要がある)。この「MP」は、Multiplan のプログラム名であり、これによってディスク上の Multiplan のプログラムがコンピュータのメモリに読み込まれて実行されて、Multiplan が立ち上がり、**図 1.3** に示すような画面が表示されます。

Microsoft®

Microsoft Multiplan
Version 3.10

ディスクからコンピュータのメモリへMultiplanのプログラムが読み込まれるとわずかな間この画面が表示され、次の表画面に変わる

Copyright (C) Microsoft Corporation, 1986, 1987. All Rights Reserved.
Microsoft is a registered trademark of Microsoft Corp.

⬇

#1　1　2　3　4　5　6　7 → 256まで

1
2
3
4
5
6
7
8
9　ユーザーはこの「電子の表」を使って、いろいろな仕事を行うことができる
10
11
12
13
14
15
16
17
18
19
↓ 4096まで

ｺﾏﾝﾄﾞ: A/文字 B/空白 C/複写 D/削除 E/修正 F/書式 G/ｾﾙ移動 H/解説 I/挿入 L/保護
M/移動 N/名前 O/ｵﾌﾟｼｮﾝ P/印刷 Q/終了 R/実行 S/並替 T/転送 V/数式 W/窓 X/連結
ｺﾏﾝﾄﾞを選択するか，英字を入力してください.
R1C1　100% Free　Multiplan: TEMP

図 1.3　自動スタートにより起動した Multiplan

以上の例は、自動スタートが設定されている Multiplan のディスクの場合ですが、もし自動スタートが設定されていない場合は、若干のキーボード操作を行わなければなりません。自動スタートでない場合の起動例を、図 1.4 に示しましょう。

```
NEC PC-9800 Series Personal Computer

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30
Copyright 1981,88 Microsoft Corp. / NEC Corporation

自由文変換システム　ＡＴＯＫ６ ver 1.2
Copyright 1986,87 (株)ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 3.30

現在の日付は 1989-07-19 (水) です.
日付を入力してください:↵
現在の時刻は 17:03:48.00 です.
時刻を入力してください:↵

A>MP↵
```

MS-DOSのオープニングメッセージ

日／時を変更しない場合は、↵のみを入力する

Multiplanのプログラムの名前「MP」を入力する

(Multiplanの起動が開始される)

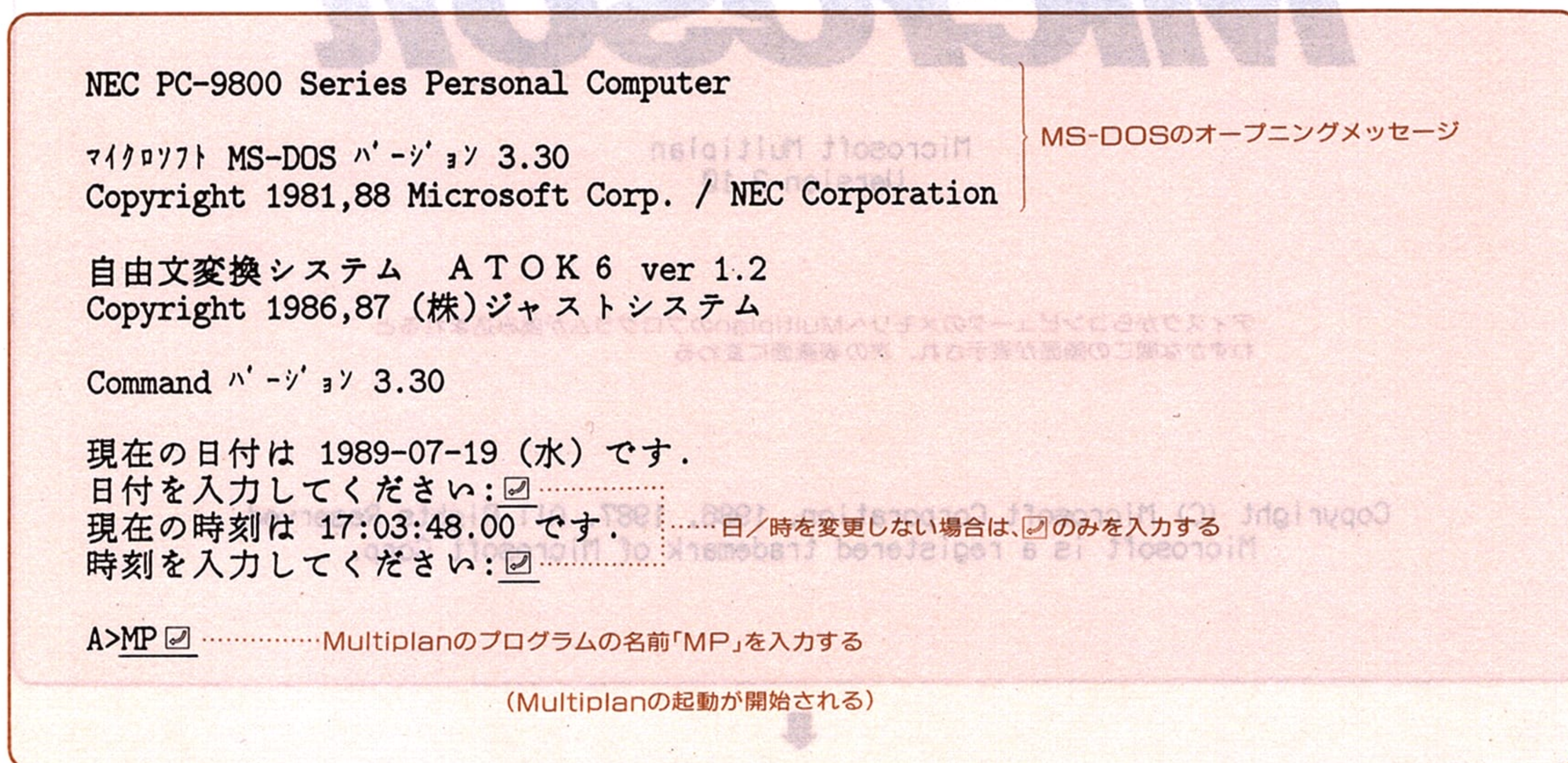



図 1.4　自動スタートでない場合の Multiplan の起動例

図 1.4 を見てもわかるとおり、自動スタートに設定されていない場合も、リセットボタン(あるいは電源 ON)により、まず MS-DOS の起動が開始され、図 1.2 の例と同様に、MS-DOS が立ち上がったことを示すオープニングメッセージが表示されます。この場合、MS-DOS はこのあと、日／時の入力を要求してきますが、表示された日／時を修正する必要がない場合は、↵(リターンキー)をキー入力します(↵の入力により日／時の入力が省略(スキップ)される)。この日／時の入力要求に答え終わると、「A＞」という記号が表示され、MS-DOS が完全に立ち上がり、私たちが MS-DOS を自由に操作できる状態になります。この状態で「A＞MP ↵」とキー入力すれば、図 1.3 の例と同じように Multiplan が立ち上がるわけです。

このように、自動スタートとそうでない場合との多少の違いはあるものの、いずれにしても、まず最初に MS-DOS が立ち上がるということがわかったと思います。(自動スタートの設定については 10 章で解説する)

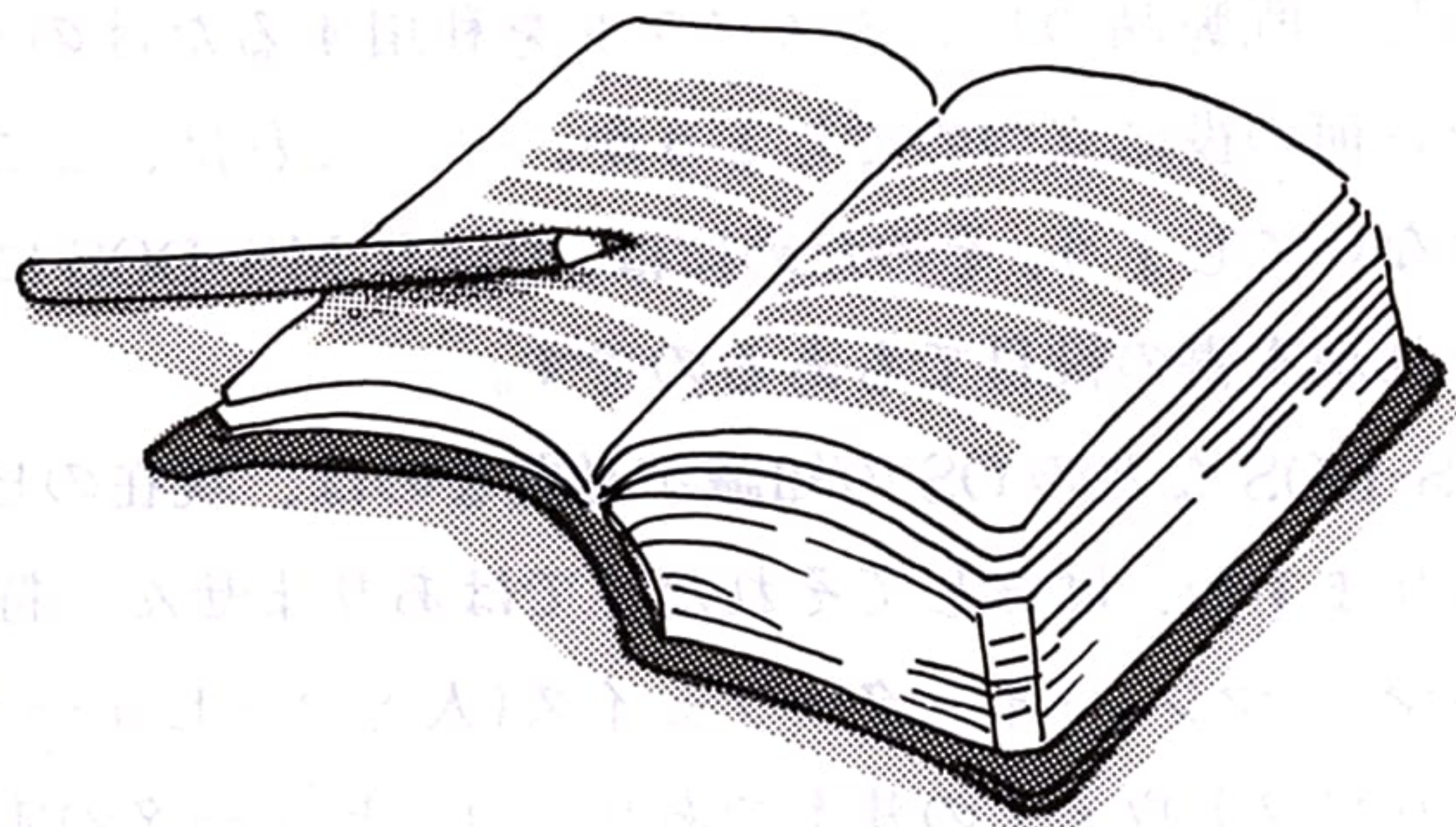
さて、起動したとしても、このようにMultiplanが立ち上がってしまえば、あとはMultiplanの世界であり、MS-DOSはMultiplanを終了するまで、表面にはほとんど顔を出しません。しかしコンピュータの内部では、MS-DOSはMultiplanを実行したり、またコンピュータの各部を動作させるために、さまざまな仕事を休みなく処理し続けているのです。

私たちはこのように、日頃使っているMultiplanやワープロソフトの一太郎などのディスクをセットして、リセットボタンを押すだけで、MS-DOSを意識することなく、それぞれのソフトの世界にはいることができます。はいってしまえばそこはMultiplanや一太郎の世界であり、表面的にはMS-DOSとは無関係です。私たちはそれぞれのソフトの世界の内部にいる限り、MS-DOSを知らなくても、あまり不自由することはないでしょう。

ではMS-DOSの知識は必要ないのでしょうか？ ―― 確かに、なくてもある程度の仕事はできます。しかし、それぞれのソフトの世界の内部に閉じこもっていたのでは、個々のソフトをいくら上手に利用できるようになっても、コンピュータの力をフルに活用することはできません。

「個々のソフトの機能」＋「MS-DOSの機能」＋「使用マシンのハードウェアの機能」

これが私たちが使うパーソナル・コンピュータの総合力であり、本格的なパーソナルコンピュータ・ユーザーは、この力をフルに活用することが必要となります。今や「BASICを学んでパソコンをビジネスに」の時代ではありません。OSを基盤にしたソフトウェアパッケージ（既成ソフトウェア商品）を利用する時代であり、そこではOSの知識が必要不可欠なのです。

本書では、そのOS（Operating System）というものの概念を知り、その機能を自由に活用できればどんな良いことがあるのかについて、1つひとつ解説していきましょう。

1.2 MS-DOSの知識はなぜ必要か

MS-DOSという名前の意味は、「マイクロソフト社ーディスク・オペレーティングシステム(MicroSoft-Disk Operating System)」です。このOSは、IBMのパーソナル・コンピュータに「PC-DOS」として搭載されているもので、16ビットマシンの標準的OSとして世界的に圧倒的なシェアを獲得し、その地位は不動のものとなりました。

「DOS」や「OS」については、現時点ではとりあえず、次のように理解しておけばよいでしょう。

DOS(ディスク・オペレーティングシステム)

パーソナル・コンピュータなど、ディスク装置を持った小規模なコンピュータシステムの基本ソフトウェアのことをいう。ディスクのファイル管理のためのプログラムは、非常に複雑で高度なものであり、OS全体でも、そのためのプログラムが占める部分が大きいので、とくにDiskの「D」を冠して呼ばれる。

OS(オペレーティングシステム)

ミニコンピュータや大型コンピュータの世界では、OSの機能が非常に拡張されているので、全体の中で、ディスク管理プログラムが占める部分は相対的に小さくなる。そこで規模の大きなコンピュータでは、DOSとは呼ばず、ただ「OS」と呼んでいる。

いずれも、コンピュータシステム全体のさまざまな基本的な機能を提供し、それらの動作を管理、統括するためのソフトウェアのことである。

MS-DOSの知識は、MS-DOS上で動作する各種のソフトウェアを開発したり、MS-DOS上の開発ツール(プログラミング言語やデバッガなど、ソフトウェアを開発するための道具としてのソフトウェア)を利用したりするプログラマーには必須であり、それに精通しなければならないのは当然です。

しかし、既製品のビジネスソフトを利用するだけの一般ユーザーにとって、MS-DOSの知識は、いったい何の役に立つのでしょうか。——これは、ここで簡単に説明しても、すぐには理解される問題ではないでしょう。そのためには、やはりMS-DOSについて1つひとつ解き明かしていく必要があり、それが本書の役目でもあるのです。

『MS-DOSなどのOSの知識が必要なのは、現在のビジネスソフトのできが悪いからだ』という意見もありますが、けっしてそれだけではありません。将来、ハード/ソフトともに発展し、さらに素晴しいマン-マシン・インターフェイス(人とコンピュータとのやり取りの部分)が実現されようと、OSはやはりソフトウェアの基本であり、コンピュータのすべての動作の基盤です。つまりコンピュータが表面的にどのように変っても、「OS」の基礎知識は、コンピュータを十分使いこなし、活用しようと

する人にとっては、すべての前提となる知識なのです。

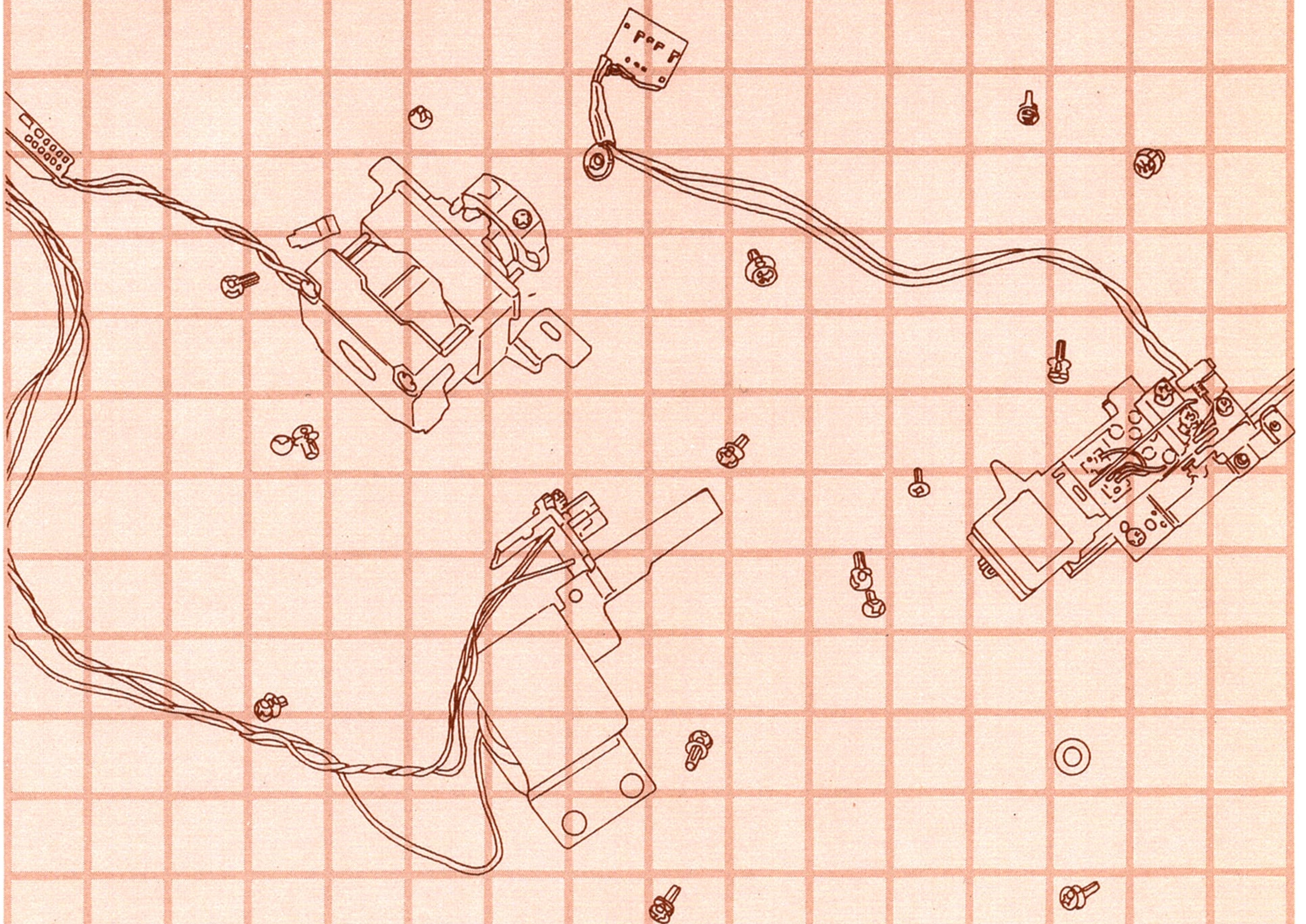
本書を読み進み、とりあえず MS-DOS の基本的な部分のいくつかが操作できるようになると、コンピュータに対する次のような点の理解が、格段に深まるでしょう。

- コンピュータの利用者にとって、重要な基礎知識である OS の概念が身につき、コンピュータ上で各種の仕事を行うことの本質的な意味が理解できるようになる。OS の概念は、BASIC をいくらやっても、また、各種のアプリケーションソフトの世界にいくら精通しても身につかないものであり、この知識なくしてパーソナル・コンピュータの本格的なユーザーにはなり得ない
- コンピュータの総合的な取り扱いが自由に行えるようになり、それぞれのアプリケーションソフトの内側だけでなく、コンピュータ全体の力(ハード＋ソフト)を利用できるようになる
- 各自が使用している個々のアプリケーションソフトの動作環境(動作状況)が理解できるので、何事にも自信を持って作業ができるようになる
- 1つのソフトの世界に閉じこもらず、各種のソフトをより広い視点で捉えることができるので、ソフトウェアを選択・採用する目ができる
- 実際の作業で重要になる「ファイル」(後述)についての知識が身につき、ディスク上のファイルの分類、整理を効果的に行うことができる
- 機種に依存しない MS-DOS ファイルの共通性、フロッピーディスクの互換性を利用して、各機種間、各種ソフト間のデータを互いに利用することが可能となる(たとえば、FM R の Multiplan で作られたデータのフロッピーディスクを、PC-9801 の一太郎に読み込んで編集し、レポートを作成するといったことなど)
- その他いろいろ

パーソナル・コンピュータのハードウェアは、その普及の歴史を築いた 8 ビットマシンから、実務機としての能力を備えた 16 ビツトマシン、さらに 32 ビットマシンへと飛躍的に発展し、とどまることがありません。

ソフトウェアの発展もめざましく、一昔前の BASIC、BASIC と騒がれていた頃に氾濫した、レベルの低いソフトとは一線を画した非常に優れたソフトが出現し、各分野の実務にどんどん使われています。それらの実務ソフトは、けっして BASIC の世界で動作するものではありません。現在のワープロソフト、表計算ソフト、データベースソフトなどの実務ソフトは、いくら優秀なプログラマーであっても、BASIC を使っていては作成することはできません。

今、私たちに要求されているのは、BASIC の知識ではなく、各種の実務ソフトを利用するための知識であり技術です。そして、その知識や技術の基本は、MS-DOS で代表される OS を学ぶことによってこそ、初めて習得できるものなのです。

2章
MS-DOSの起動

MS-DOSや各種のソフトが実行される以前、それぞれのプログラムはディスク上の「プログラムファイル」として存在しています。それらをコンピュータ本体のメモリ上に読み込んで実行することにより、各プログラムの目的の動作が開始されますが、そのことを、「起動」とか「立ち上げ」「スタートアップ」などと呼んでいます。前章では、ビジネスソフトを立ち上げる過程における、MS-DOSの起動について触れましたが、ここでは、「MS-DOSの起動」そのものに注目して解説しましょう。

ここでの実習には、MS-DOSのシステムディスクを使います。もしそれが手元にない場合は、Multiplanや一太郎などの、日常使っている各種のソフトのシステムディスクに含まれるMS-DOSを利用してください。

MS-DOSが起動したあとは、その次のステップとして、MS-DOSの多くの機能の中でも、最もポピュラーな、ディスクの内容を見るための「DIR」というコマンドを実行し、ディスク上にどのようなファイルが存在しているかを見てみましょう。

2.1 MS-DOSを起動する前に

私たちがビジネスソフトを利用したり、ソフトウェアの開発を行ったりする場合の、コンピュータシステムの状況 —— つまり、

①私たち**ユーザー**
②コンピュータの本体(**ハードウェア**)
③ **MS-DOS**
④**アプリケーションソフト**(Multiplanや一太郎などのビジネスソフトや、その他さまざまな、私たちがOS上で利用するソフトウェアのこと。「ユーザープログラム」とも呼ばれる)

などの関係を、まず簡単に説明しておきましょう(図2.1)。とくにコンピュータシステムの中でのMS-DOSの位置、立場に注目してください。

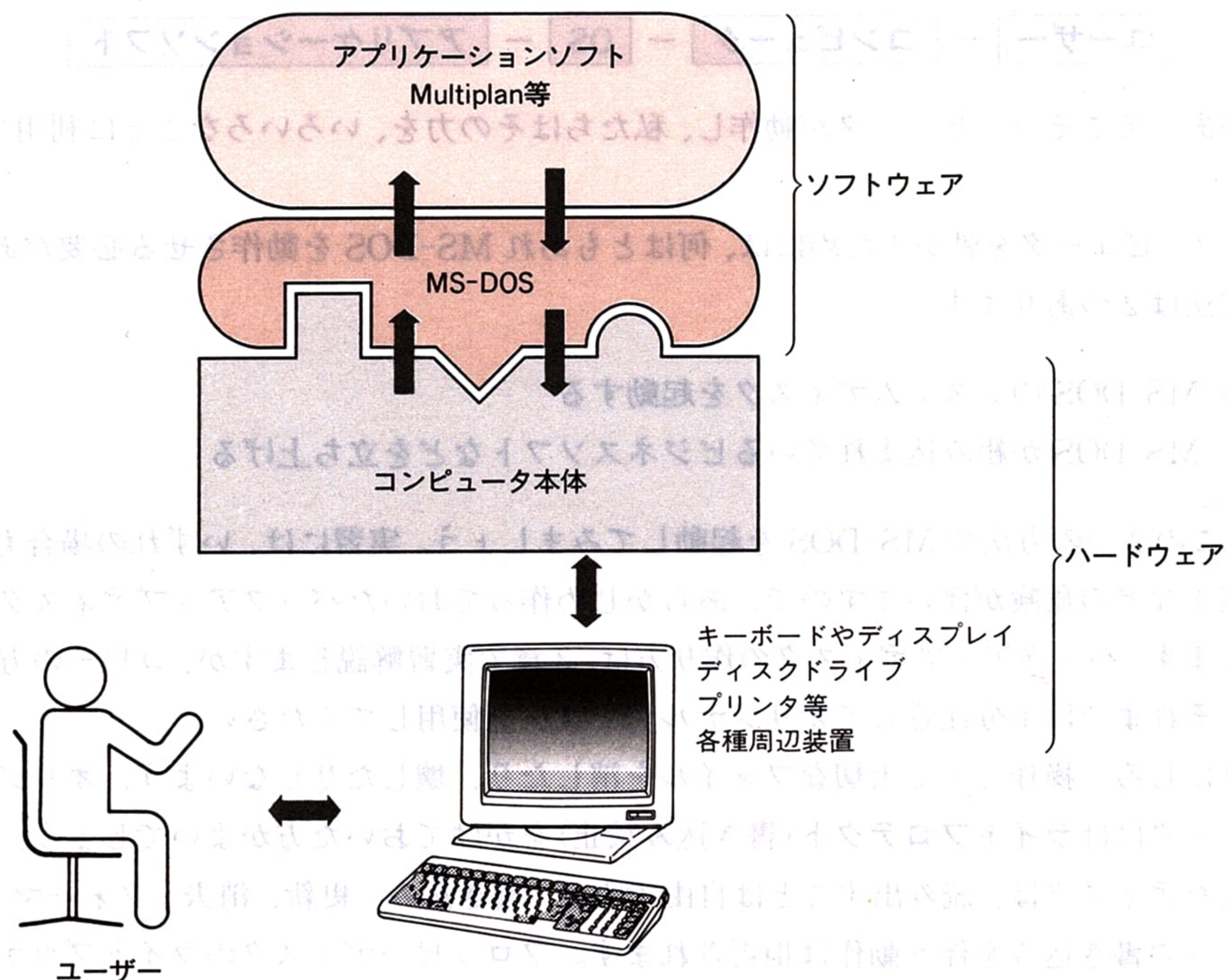



図2.1 動作しているコンピュータの中でのMS-DOSの位置

図2.1は次のように理解してください。

まずコンピュータは、みなさんもよくご承知のように「ソフトがなければただの箱」ですので、コンピュータそのもの(ハードウェア)だけではまったく動作しません。電源を入れただけでは、なにもできないコンピュータを動作させるには、ソフトウェアが必要です。しかし、Multiplanや一太郎などのアプリケーションソフトを直接コンピュータに与えても、これまたコンピュータはなにもできないのです。ここは非常に大切なところです。

では、コンピュータはどうすれば動作するのでしょうか。

コンピュータを動かすためには、まずコンピュータの基本的な動作を行わせるためのソフトウェアを与える必要があります。このソフトウェアがMS-DOSやCP／MやMS-OS/2などの「基本ソフトウェア」(OS：オペレーティングシステム)なのです。

「コンピュータ＋MS-DOS」によって、コンピュータシステムは、アプリケーションソフトを実行することが可能になります。私たちはMS-DOSの上で、初めてビジネスソフトやプログラミング言語などの各種のソフトウェアを実行し、それぞれの仕事を行わせることができるのです。図2.1は、そのような状況や意味を表しています。

大型コンピュータの世界ではあたり前のことですが、パーソナル・コンピュータにおいても、

ユーザー － **コンピュータ** － **OS** － **アプリケーションソフト**

の関係があってこそコンピュータが動作し、私たちはその力を、いろいろなことに利用できるのです。

さて、コンピュータを動かすためには、何はともあれMS-DOSを動作させる必要があります。そのための方法は2つあります。

a) MS-DOSのシステムディスクを起動する
b) MS-DOSが組み込まれているビジネスソフトなどを立ち上げる

では、この2つの方法でMS-DOSを起動してみましょう。実習には、いずれの場合も、ファイルを誤って壊すなどの危険が伴いますので、あらかじめ作っておいたバックアップディスクを使うことをお勧めします。バックアップディスクの作り方は、3章で実習解説しますが、コピーの方法がわからない方は、それまでは十分注意してオリジナルディスクを使用してください。

いずれにしろ、操作ミスで大切なファイルを消したり、壊したりしないよう、オリジナルのフロッピーディスクには**ライトプロテクト**(書き込み禁止)をかけておいた方がよいでしょう。ライトプロテクトされたディスクは、読み出すことは自由ですが、書き込み、更新、消去、フォーマット処理など、ディスクへの書き込みを伴う動作は拒否されます。フロッピーディスクのライトプロテクトの方法を**図2.2**に示します。

3.5インチディスクの場合

5インチディスクの場合

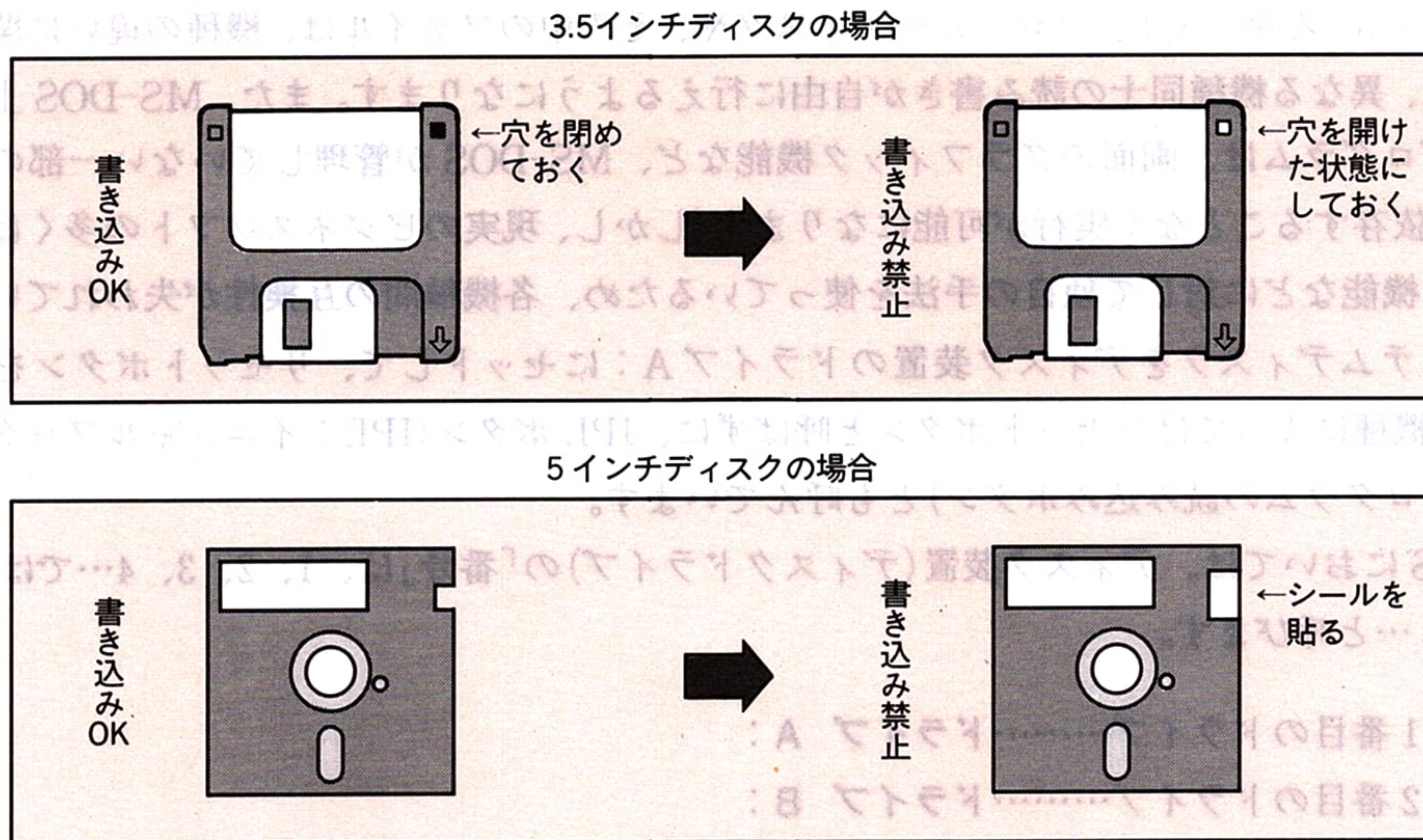



図 2.2　フロッピーディスクのライトプロテクトの方法

2.2　話はあとから、まず起動

MS-DOSのシステムディスクが手元にない方も、最初の「MS-DOSシステムディスクによる起動」からお読みください。

■ MS-DOSシステムディスクによる起動

MS-DOSのシステムディスクは、MS-DOSが組み込まれていないビジネスソフトや、その他のさまざまなアプリケーションソフトを実行したりする場合に必要となります。どのような仕事をする場合でも、まず最初に、基本ソフトウェアであるMS-DOSを起動しておかなければなりません。つまり、コンピュータシステム全体を、基本的動作が可能な状態にしておくわけです。

MS-DOSのシステムディスクは、8086やその相当品のV30、さらに80286等の16ビットCPUを使った各社のパーソナル・コンピュータの機種ごとに、それぞれのメーカーから提供されています。購入の際には、各自が使っている機種用のもので、かつ提供されるフロッピーディスクのタイプ(5インチ2HDとか、3.5インチ2HDとかの種類)が、使っているフロッピーディスク装置に合ったものを選ばなければなりません。

しかし、いずれの機種も、いったんMS-DOSが起動すると、コンピュータは全機種共通の「MS-DOSマシン」に変身します。フロッピーディスクや、その中のファイルは、機種の違いに関係なく互換性があり、異なる機種同士の読み書きが自由に行えるようになります。また、MS-DOS上で実行する各種のプログラムは、画面のグラフィック機能など、MS-DOSが管理していない一部の機能を除き、機種に依存することなく実行が可能になります(しかし、現実のビジネスソフトの多くは画面のグラフィック機能などに対して独自の手法を使っているため、各機種間の互換性が失われている)。

このシステムディスクをディスク装置のドライブA：にセットして、リセットボタンを押します(**図2.3**)。機種によってはリセットボタンと呼ばずに、IPLボタン(IPL：イニシャルプログラムローダ。初期プログラムの読み込みボタン)とも呼んでいます。

MS-DOSにおいては、ディスク装置(ディスクドライブ)の「番号」は、1、2、3、4…ではなく、A、B、C、D、…と呼びます。

1番目のドライブ………**ドライブ　A：**
2番目のドライブ………**ドライブ　B：**
3番目のドライブ………**ドライブ　C：**
4番目のドライブ………**ドライブ　D：**

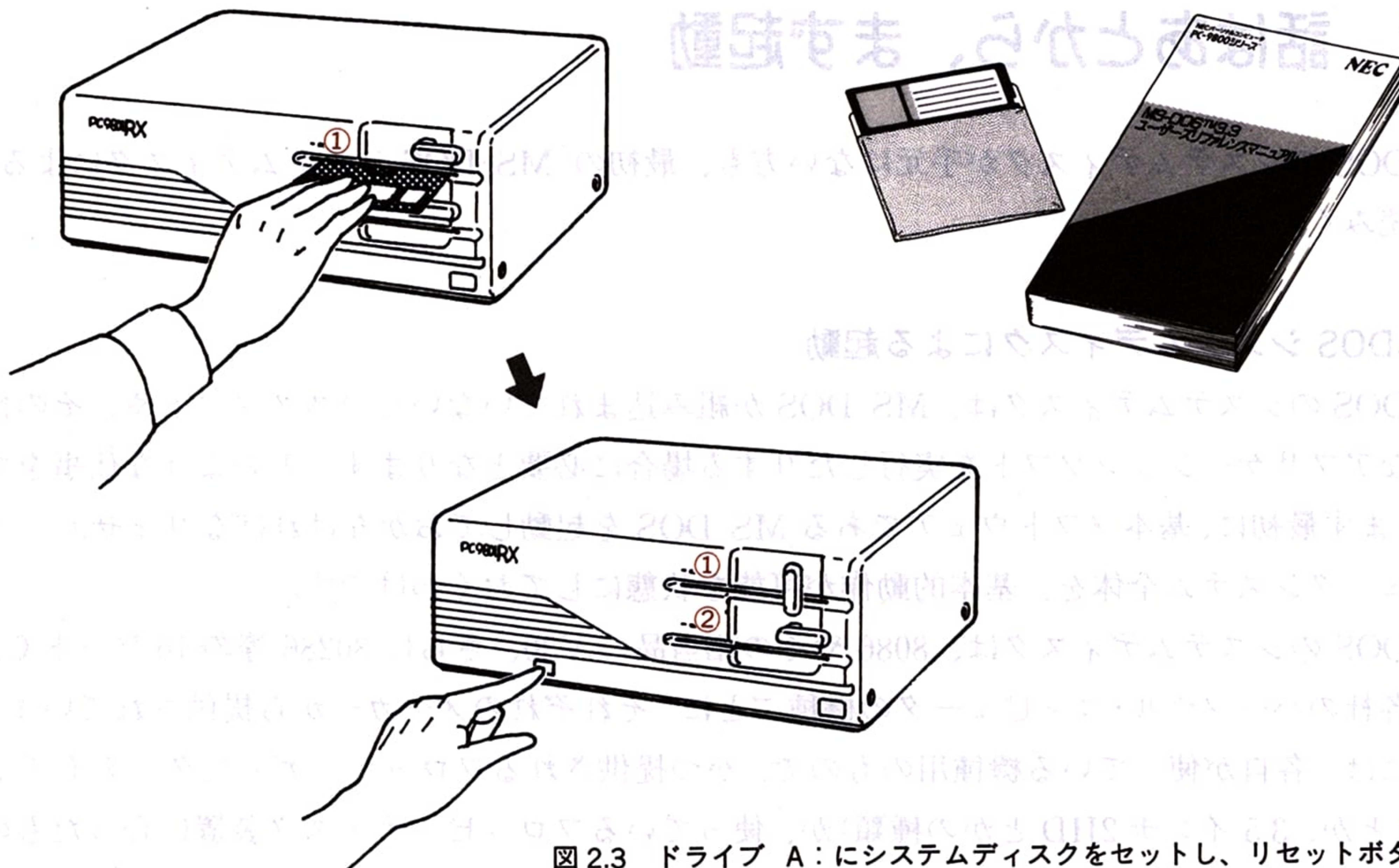



図2.3　ドライブ　A：にシステムディスクをセットし、リセットボタンを押す

MS-DOSのシステムディスクをセットしてリセットボタンを押すと、コンピュータ本体のROM(読み出し専用のメモリ)に固定的に書き込まれている、起動のためのプログラム(**ブートローダ**と呼ばれる)が働き出し、ディスクの最外周の最初の部分に書き込まれている1つの小さなプログラムを読み出し、それを実行します。このプログラムは、MS-DOSを起動するためのプログラムであり、それが実行されることによって、いよいよMS-DOS本体のプログラムがディスクから読み出され、そのプログラムが実行されてMS-DOSが立ち上がります。

このようにしてMS-DOSが立ち上がると、MS-DOSは最初、図2.4のように日付と時刻の入力を要求してきます。

```
NEC PC-9800 Series Personal Computer

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30
Copyright 1981,88 Microsoft Corp. / NEC Corporation

  プリンタが使用可能です

  ＲＳ－２３２Ｃインターフェイスが使用可能です

  ＡＩかな漢字変換が使用可能です
  辞書は、ドライブ B: の NECAI   .SYS です

Command ﾊﾞｰｼﾞｮﾝ 3.30

現在の日付は 1989-07-19 (水) です.
日付を入力してください:↵
現在の時刻は 17:03:48.00 です.
時刻を入力してください:↵

A>
```

これらのメッセージは、それぞれのシステムディスクの初期設定の状況によって異なる

変更の必要がない場合は↵を入力する

図2.4　MS-DOS起動時における日／時の入力要求

MS-DOSはこのように、コンピュータの内部に組み込まれている「カレンダー時計」が保持している日付と時刻を表示してから、新たな日／時の入力を要求してきます。表示された日／時が正しければ変更する必要はありませんので、その場合は図2.4の例のように↵を入力し、要求を無視します。変更する場合は、図2.5に示すように、表示された形式と同じ形式で、正しい日／時をキー入力します。たいていのコンピュータは、内部のカレンダー時計の回路を、充電式の電池でバックアップしていますので、長期間電源をOFFにしておかない限り、カレンダー時計の動作は停止することはありません。

日／時の入力が済むと、「A＞」という記号が表示されます。これでMS-DOSが完全に立ち上がりました。このときのMS-DOSは、私たちが望むさまざまな処理を行わせるための各種の**コマンド**(命令)を待ち受ける状態にあります。

```
Command ﾊﾞｰｼﾞｮﾝ 3.30

A>DATE
現在の日付は 1989-07-19 (水) です.
日付を入力してください: 2001-4-1⏎ ……………2001年4月1日を入力する

A>TIME
現在の時刻は 17:03:48.00 です.
時刻を入力してください: 12:30:00⏎ ……………12時30分00秒を入力する

A> ……………完全に起動したMS-DOSのプロンプト
```

図 2.5　日／時を変更する場合のキー入力と、その後完全に立ち上がったMS-DOS

さて、図2.4、図2.5の最後に表示されている「A＞」に注目してください。この「A＞」こそMS-DOSが完全に立ち上がったことを示す記号で、**プロンプト**と呼ばれるものです(機種によっては、「＞」の部分にほかの記号が使われているものもある)。「プロンプト」とは、コマンドの入力をうながすという意味であり、MS-DOSが、私たちの要求するコマンドを受け付け可能な状態にあることを知らせるものです。

「A＞」のAは、現在、ドライブA：が「メインのディスク」となっていることを表すもので、この部分は、その後の操作によって、「B＞」とか「C＞」とかが表示される場合もあります。しかしいずれの場合も、MS-DOSの「プロンプト」であることに変わりはありません(「メインのディスク」について詳しくは、7.3節で解説する)。

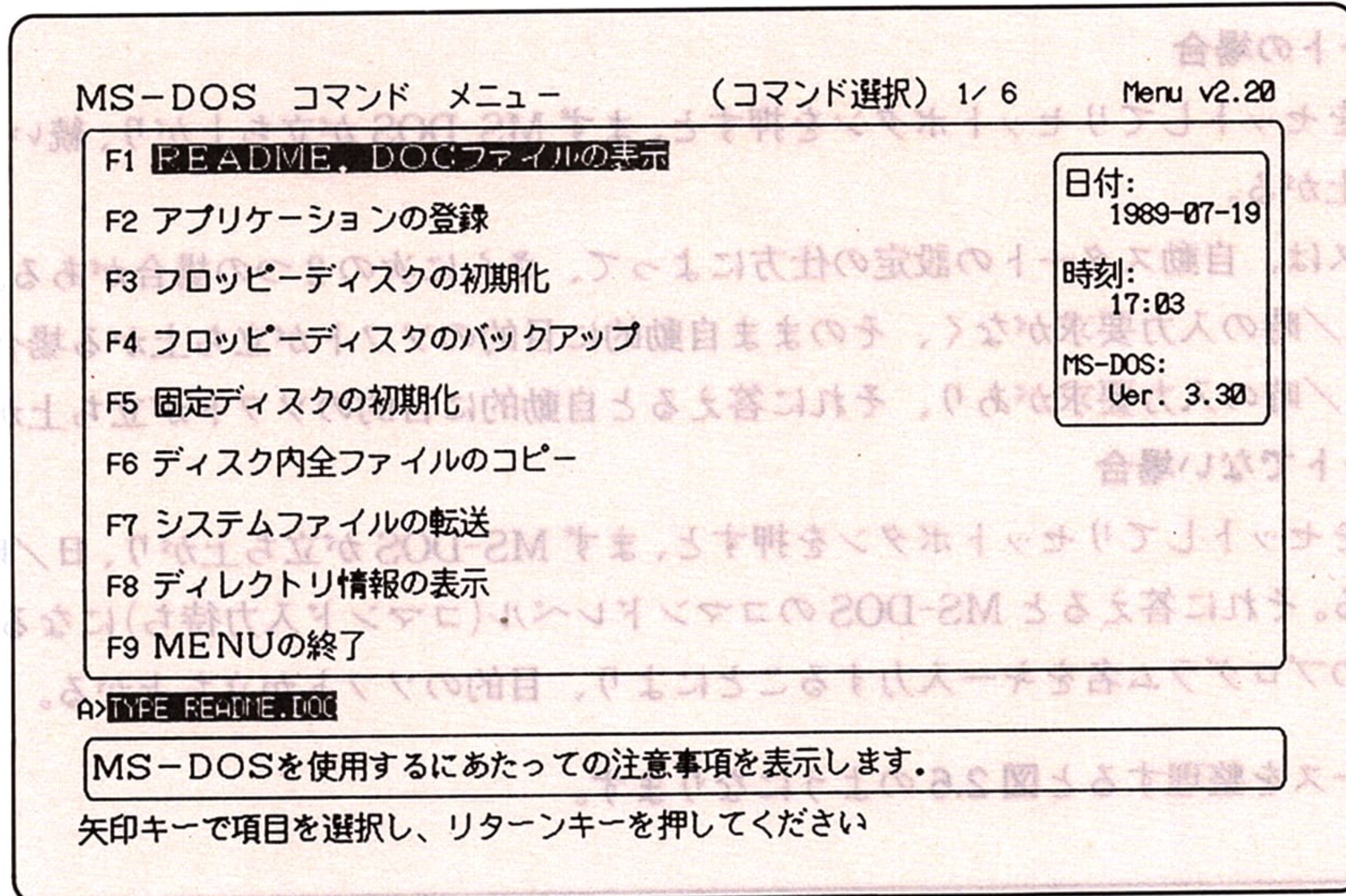
なお、この「A＞」が表示されている状態のことを、MS-DOSの**コマンドレベル**と呼びます。一般にもよく使われる言葉ですので、覚えておいてください。

MS-DOSの起動についてもう少し正確にいうと、実はMS-DOSは、日／時の入力を要求する以前にすでに立ち上がり、動作を開始しているのです。立ち上ったあと、ユーザーに対する最初の動作として日／時の入力を要求し(7.10節で解説するDATEとTIMEコマンドをMS-DOS自身が実行する)、その入力の処理が済むとコマンドレベルを示すプロンプト「A＞」を表示するわけです。

以上で、MS-DOSのシステムディスクによって、MS-DOSが完全に起動しました。MS-DOSのシステムディスクを使って実習している方は、2.3節にジャンプしてもかまいません。

MS-DOS のコマンドメニュー

MS-DOS の起動は、本来は、さきの図 2.4 のような形式で立ち上がりますが、各メーカーのシステムディスクによっては、それぞれのメーカーが独自に作った「コマンドメニュー」が表示されるかも知れません。下の図は、PC-9800 用の MS-DOS システムディスク(バージョン 3.1 や 3.3)を起動した際に、自動的に表示されるコマンドメニューです。

```
MS-DOS  コマンド  メニュー          (コマンド選択) 1/ 6          Menu v2.20
 F1 README.DOCファイルの表示                          日付:
 F2 アプリケーションの登録                               1989-07-19
 F3 フロッピーディスクの初期化                          時刻:
 F4 フロッピーディスクのバックアップ                       17:03
 F5 固定ディスクの初期化                                MS-DOS:
 F6 ディスク内全ファイルのコピー                           Ver. 3.30
 F7 システムファイルの転送
 F8 ディレクトリ情報の表示
 F9 MENUの終了
A>TYPE README.DOC
MS-DOSを使用するにあたっての注意事項を表示します.
矢印キーで項目を選択し、リターンキーを押してください
```

このようなコマンドメニューは、本来の MS-DOS とは、なんの関係もありませんので、本書では無視してください。これらは、「MS-DOS の操作が楽になる、と思って各メーカーが勝手に作った1つのプログラム」にすぎません。このようなコマンドメニューが表示されるのは、私たちが日常使っているビジネスソフトを自動スタートする場合と同じ手法で、そのプログラム(MENU.COM)が MS-DOS の起動時に自動スタートするようになっているからです。マン-マシン・インターフェイスとしてもっと洗練されたコマンドメニューなら話は別ですが、MS-DOS を学ぼうとする人にとって、このようなコマンドメニューの存在は、かえって「じゃま」になるだけでしょう。MS-DOS の本来の姿を見ようとする人にとっては、混乱を招くだけです。

もしこのようなコマンドメニューが表示されてしまった場合は、すぐさま「終了」を選択してください。このようなメニューが自動的に表示されない本来の MS-DOS の姿に戻すには、自動スタートを指示する部分から、コマンドメニュープログラムの箇所を削除すればよいわけですが、自動スタートの設定や解除に関しては、10 章で解説していますので、とりあえずはすぐさま「終了」を選択し、本来の MS-DOS の「A＞」が表示されればよいとしましょう。

繰返しますが、このようなコマンドメニューは、MS-DOS 上で動作する一般のプログラムの1つであり、『本来の MS-DOS とはなんの関係もない』ということをはっきり認識しておいてください。

■ ビジネスソフトからの起動

それぞれのメーカーから提供されている MS-DOS のシステムディスクが手元にない方は、日常使っている各種のビジネスソフトなどのシステムディスクに組み込まれている MS-DOS を利用します。

MS-DOS が組み込まれている各種のソフトの起動には、自動スタートの設定がしてあるか／ないかによって、次の 2 つの場合があります。

(1) 自動スタートの場合

ディスクをセットしてリセットボタンを押すと、まず MS-DOS が立ち上がり、続いて目的のソフトが立ち上がる。

このケースは、自動スタートの設定の仕方によって、さらに次の 2 つの場合がある。

(a) 日／時の入力要求がなく、そのまま自動的に目的のソフトが立ち上がる場合

(b) 日／時の入力要求があり、それに答えると自動的に目的のソフトが立ち上がる場合

(2) 自動スタートでない場合

ディスクをセットしてリセットボタンを押すと、まず MS-DOS が立ち上がり、日／時の入力を要求してくる。それに答えると MS-DOS のコマンドレベル(コマンド入力待ち)になるので、実行するソフトのプログラム名をキー入力することにより、目的のソフトが立ち上がる。

これらのケースを整理すると**図 2.6** のようになります。

(1)(a) 自動スタートの場合
リセットボタンを押す
(1)(b) 自動スタートの場合
コマンドレベル
MS-DOSが起動する
日／時の入力要求
MS-DOSのプロンプト A＞
ビジネスソフト等の起動
目的のソフトの実行
自動スタートでない(2)の場合
(日／時あるいは⏎の入力)
(CTRL ＋ C の入力により、(1)(b)のケースであってもコマンドレベルになる)
(各種のコマンドや実行するプログラムの入力)
(各ソフトの「終了」コマンドにより、MS-DOS のコマンドレベルに戻る)

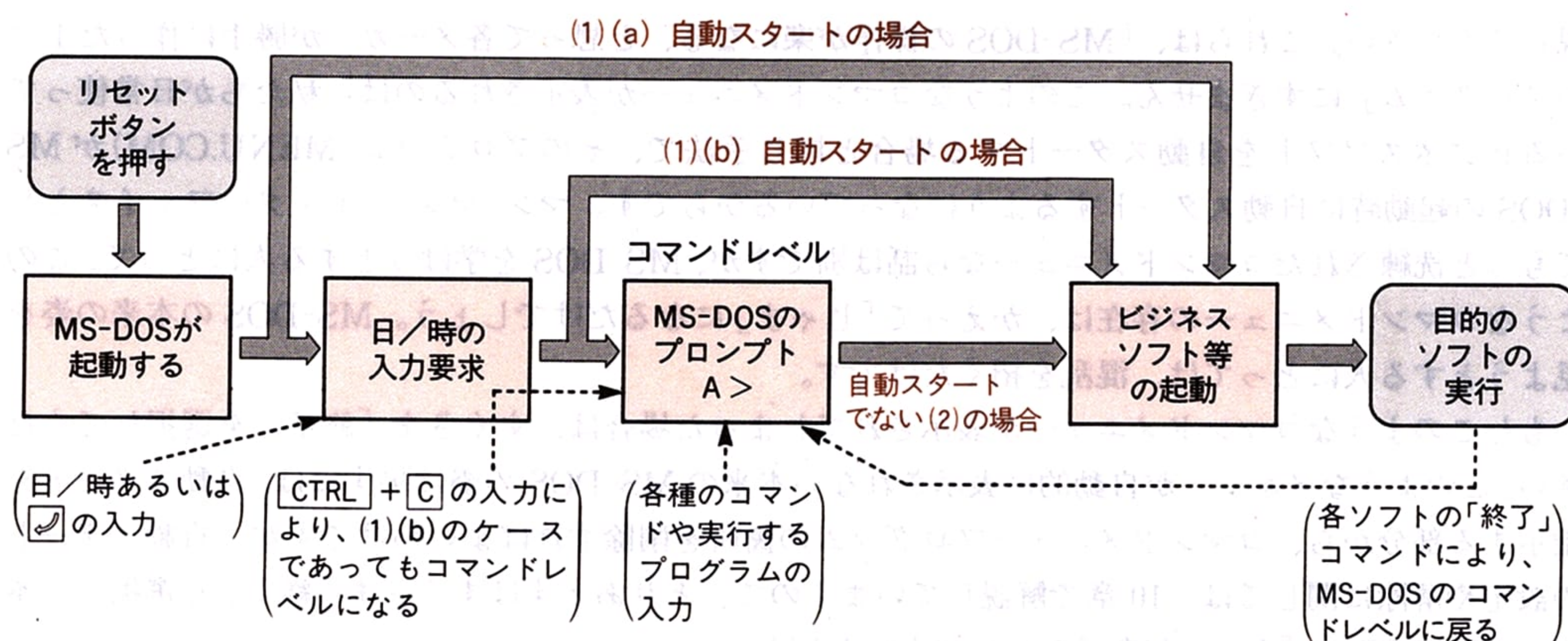



図 2.6　MS-DOS を組み込んだ各種ビジネスソフトの起動法

ではここで、いくつかの実際の起動例を示しましょう。いろいろな状態に設定されているビジネスソフトのディスクをドライブA：にセットしてリセットボタンを押してみましょう。いずれの場合も、最終的にはMS-DOSのプロンプト「A>」が表示されればOKです。

ビジネスソフトが立ち上がってしまった場合

ビジネスソフトのディスクをセットして、リセットボタンを押すことにより、自動的にそのビジネスソフトが立ち上がってしまった場合、ユーザーは、MS-DOSの世界ではなく、いきなりそれぞれのソフトの世界にはいってしまいます。

その場合、MS-DOSのコマンドレベルに下りるには、そのソフトを終了させる必要があります。たいていのビジネスソフトは、その実行を「終了」させ、MS-DOSに下りる機能を持っています。ここでは、そのMultiplanの例を示しておきましょう（図2.7）。

```
#1    1        2        3        4        5        6        7
  1
  2
  3
  4
  5
  6
 13
 14
 15
 16
 17
 18
 19
ｺﾏﾝﾄﾞ: A/文字 B/空白 C/複写 D/削除 E/修正 F/書式 G/ｾﾙ移動 H/解説 I/挿入 L/保護
  M/移動 N/名前 O/ｵﾌﾟｼｮﾝ P/印刷 Q/終了 R/実行 S/並替 T/転送 V/数式 W/窓 X/連結
ｺﾏﾝﾄﾞを選択するか，英字を入力してください．
R1C1                                    100% Free     Multiplan: TEMP
```

終了コマンドを選択し⏎を入力する

```
A> ……………Multiplanを終了して、MS-DOSのコマンドレベルに戻った
```

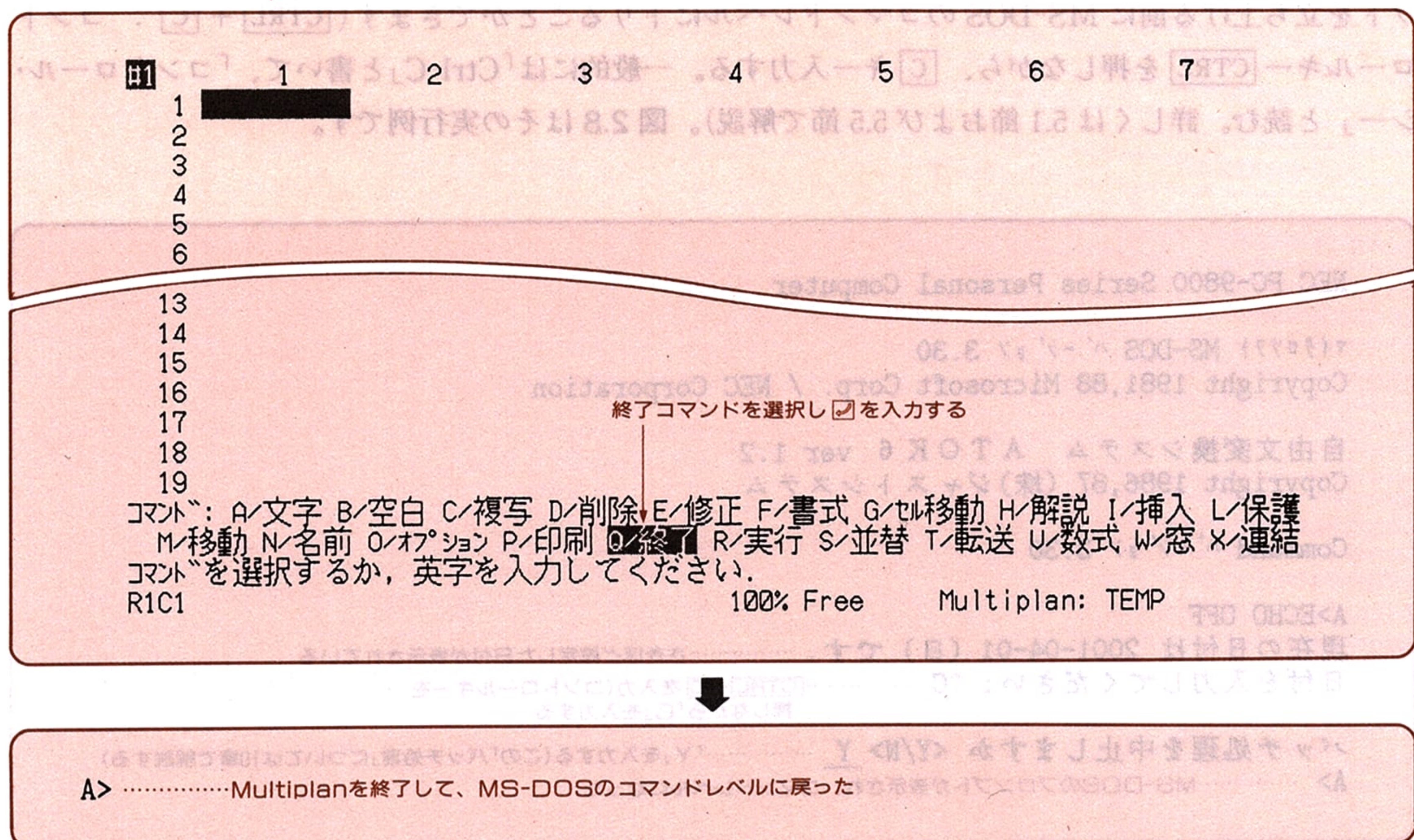



図2.7　Multiplanの終了コマンドを選択し、MS-DOSのコマンドレベルに下りた例

ソフトによって操作法は異なりますが、それぞれMS-DOSのコマンドレベルに下りるための「終了」コマンドが用意されています。

実習されている方で、ここで無事にプロンプト「A>」が表示された場合は、とりあえず次項はパスして、2.3節に進んでもかまいません。

日／時を聞いてきた場合

ビジネスソフトを起動する際、日／時の入力を要求してきた場合は、要求どおりに日／時を入力するか、もしくは日／時の入力をスキップ(無視)する意味の⏎を入力します。そのあとは、MS-DOSのコマンドレベルとなるか、あるいは自動スタートがかかって、それぞれのソフトが立ち上がってしまうかのいずれかになります。前者は自動スタートがまったく設定されていない場合((2)のケース)であり、後者はDATEコマンドとTIMEコマンド、さらに、そのあとそれぞれのビジネスソフトのプログラムが自動スタートとして設定されている場合((1)bのケース)です。

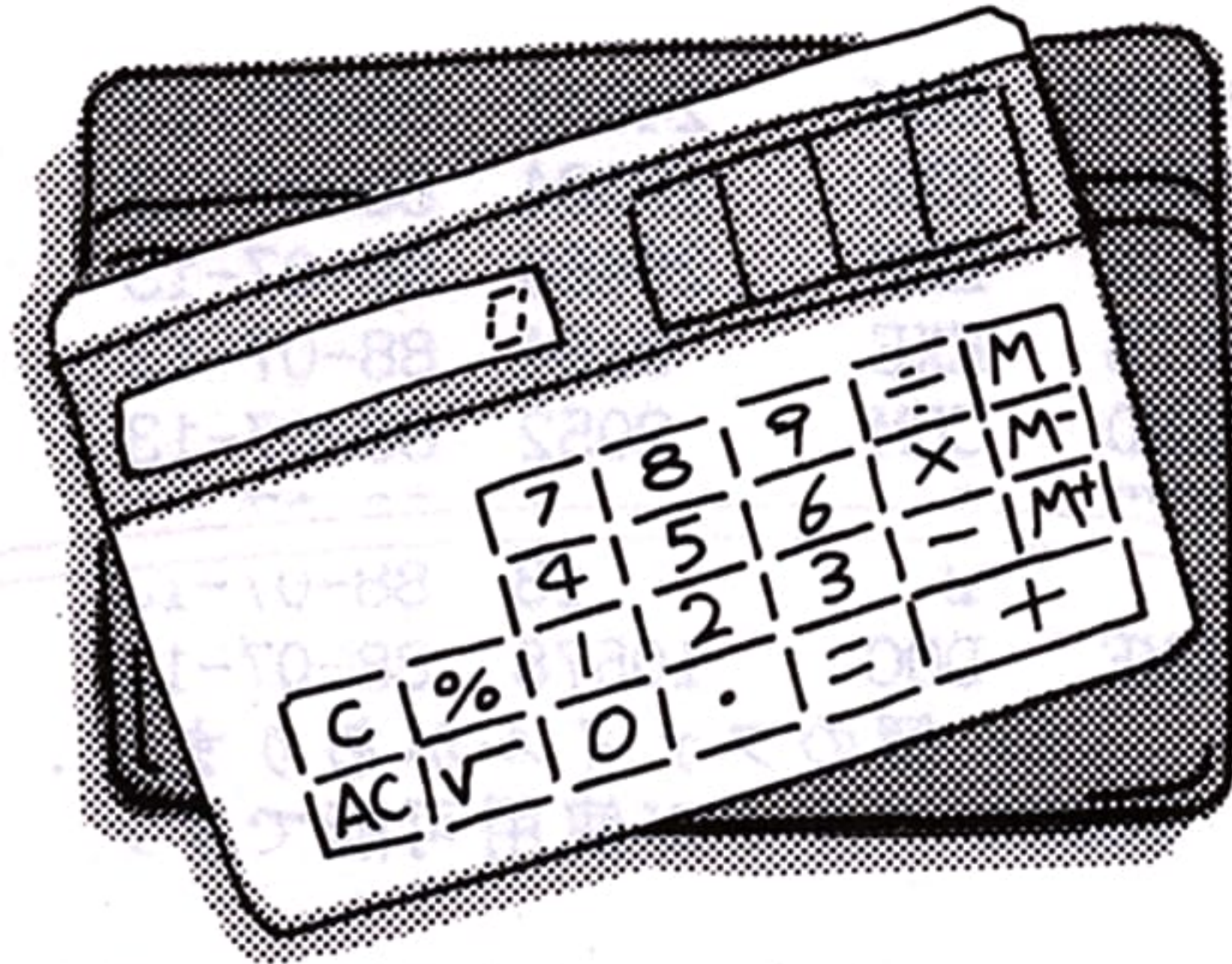
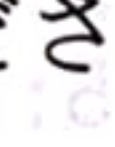
後者の場合は、日時の入力要求に答えると、自動的にそれぞれのソフトが立ち上がってしまいます。その場合、前項のようにいったん立ち上げてからそのソフトを「終了」し、MS-DOSのコマンドレベルに下りてもよいのですが、日／時の入力待ちの状態からは、もっと簡単にMS-DOSのコマンドレベルに下りることができます。

その操作は簡単であり、日／時の入力要求に対して、[CTRL]＋[C]をキー入力することにより、各ソフトを立ち上げる前にMS-DOSのコマンドレベルに下りることができます([CTRL]＋[C]： コントロールキー[CTRL]を押しながら、[C]キー入力する。一般的には「Ctrl-C」と書いて、「コントロール・シー」と読む。詳しくは5.1節および5.5節で解説)。**図2.8**はその実行例です。

```
NEC PC-9800 Series Personal Computer

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30
Copyright 1981,88 Microsoft Corp. / NEC Corporation

自由文変換システム　ＡＴＯＫ６　ver 1.2
Copyright 1986,87 (株)ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 3.30

A>ECHO OFF
現在の日付は 2001-04-01 (日) です. ……………さきほど設定した日付が表示されている
日付を入力してください: ^C ……………[CTRL]+[C]を入力(コントロールキーを
                                       押しながら「C」を入力する
バッチ処理を中止しますか <Y/N> Y ……………「Y」を入力する(この「バッチ処理」については10章で解説する)
A> ……………MS-DOSのプロンプトが表示され、コマンドレベルになった
```

図2.8　日／時の入力要求に対して[CTRL]＋[C]をキー入力することによりMS-DOSに下りることができる

前節でも触れましたが、日／時の入力要求は、MS-DOSがDATEとTIMEコマンド(7.10節で解説する)を実行しているために発生するものですので、そのコマンドの実行を[CTRL]＋[C]でキャンセルするわけです。

さて、みなさんは、何らかの方法でMS-DOSを起動して、MS-DOSの世界にはいること――つまりMS-DOSのプロンプト「A>」を表示することができたと思います。次節からはいよいよMS-DOSを操作してみましょう。

2.3　キー入力とミスタイプの修正

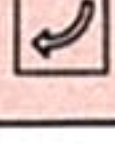
みなさんの前の画面に、MS-DOSのプロンプト「A>」と、その右にカーソルが表示されているとしましょう(図2.9)。

```
A>■
```

図2.9　MS-DOSのすべては、このプロンプトの画面から始まる

では、MS-DOSのコマンドの中から、簡単なものをいくつか実際に操作して、その機能を見ていきましょう。この「A>」の状態を、MS-DOSの「コマンドレベル」と呼ぶことは2.2節で述べました。コマンドレベルとは、ユーザーがキー入力するさまざまな命令(コマンド)を受け付け、そのコマンドの文字列がどのような命令を意味するかを識別し、その命令を実行するためのすべての用意が整っている状態のことをいいます。

何はともあれ、まず「DIR↵」と入力してみてください。文字は大文字、小文字のどちらを使ってもかまいません。プロンプト「A>」のあとに続けて、

A>DIR↵

とキー入力します。(図2.10。図中の下線部はユーザーがキー入力する部分、↵はリターンキーの入力を示す)

```
A>DIR⏎ ・・・・・・・・・・DIRコマンドを実行する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13    0:00
ADDDRV   EXE    18384  88-07-13    0:00
ASSIGN   EXE    26450  88-07-13    0:00
ATTRIB   EXE     9154  88-07-13    0:00
APPEND   COM     2052  88-07-13    0:00
README   DOC    10578  88-07-13    0:00
        57 個のファイルがあります.
     26624 バイトが使用可能です.

A> ・・・・・・・・・・DIRコマンドの実行が終わって、再びMS-DOSのプロンプトに戻っている
```

(3枚組のシステムディスクの中の＃1、「ブートディスク」と呼ばれているものの内容)

表示が１画面におさまらない場合は、CTRL＋S（CTRLキーを押しながらSを押す）で表示を一時ストップさせることができる。何らかのキー入力でストップは解除される(5.4節で詳しく解説する)

図 2.10　PC-9800 シリーズの MS-DOS システムディスクに対する DIR コマンドの実行例

このように、いくつかのファイルの名前やその日付などが表示されました。表示される内容は、セットされているディスクによって異なりますが、この例は、PC-9800 シリーズの MS-DOS のシステムディスクの場合です。

DIR コマンドのもう１つの実行例として、MS-DOS が組み込まれたビジネスソフト Multiplan のディスクの場合を示しておきましょう(図 2.11)。

```
A>DIR⏎ ・・・・・・・・・・Multiplanのディスクをセットして、DIRコマンドを実行する

 ドライブ A: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13    0:00
FORMAT   EXE    97766  88-07-13    0:00
MP       COM    36210  87-07-12    3:10
MP       COD    71904  87-07-12    3:10
AUTOEXEC BAT       27  89-07-19   15:12
MP       INI      258  89-07-19   17:24
        24 個のファイルがあります.
     34816 バイトが使用可能です.

A> ・・・・・・・・・・実行が終わり、再びMS-DOSのコマンドレベルに戻っている
```

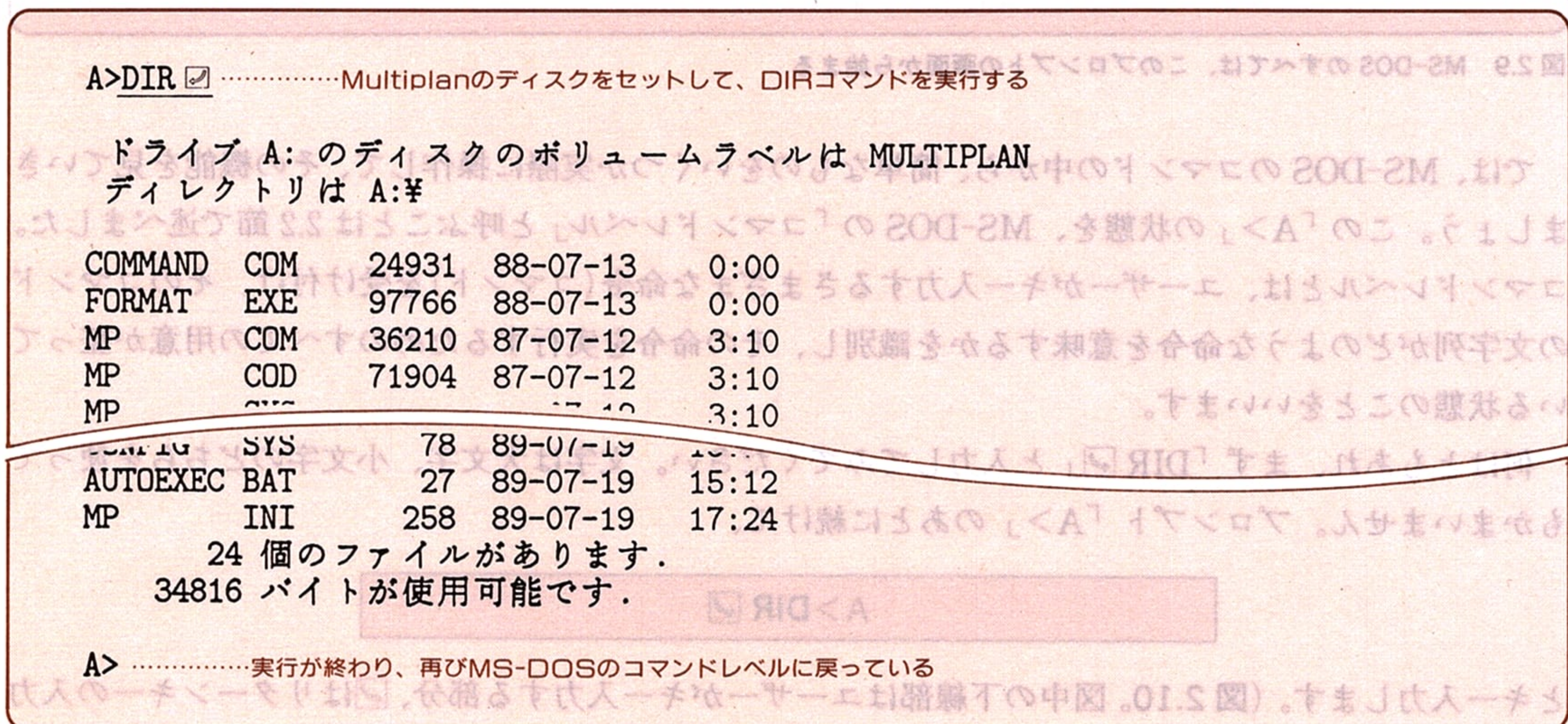



図 2.11　Multiplan のディスクの DIR コマンドの実行例

いずれにしろ、DIRというコマンドの実行によって、画面には、そのディスクの内容が表示されたと思います。このとき、MS-DOSの**コマンド・プロセッサ**と呼ばれる部分 —— 入力された命令(コマンド)を識別しその実行を処理する部分が、キー入力された「D」「I」「R」という文字列の意味を識別し、そのDIRコマンドのプログラム(MS-DOSに内蔵されている)を実行したわけです。

もし私たちが入力したコマンドが、MS-DOSに内蔵されていないプログラムで、かつ、ディスク上にもないプログラムであった場合は、実行不可能であることを知らせる**エラーメッセージ**を表示してコマンドレベルに戻ります。その例を**図2.12**に示します。(コマンドやプログラムの実行について詳しくは8章で解説する)

```
A>ABCDEF ⏎ ……………でたらめな名前のコマンドやプログラムを実行する
コマンドまたはファイル名が違います

A>
```

図2.12　存在しないコマンドやプログラム名を入力した場合のMS-DOSのエラーメッセージ

さて、さまざまなコマンドの入力の際、ミスタイプしてしまった場合は、[BS](Back Space)キー、もしくは[DEL](DELete)キーで1文字ずつ削除することができます。これは簡単ですので、実行例は示さなくてもよいでしょう。[BS]キーと[DEL]キーは、本来機能が異なるのですが、MS-DOSのコマンドレベルでは、どちらもカーソルの左側の1文字を削除するという、同じ機能となります。

MS-DOSには、「テンプレート機能」と呼ばれる、入力したコマンドラインに対する簡単な編集機能がありますが、たいへん操作性の悪いものであり、本書では触れません。この操作は、BASICやワープロで文字を書くときのように、ミスタイプしたところへカーソルを移動して書き直したり、カーソルで指定したところの文字を削除するなどの自由な操作ができない不便なものですが、同じコマンドラインを再入力する機能など、まあ知っていた方がよいものも1つ2つありますので、何かの機会があったら調べてみたらいかがでしょう。

もし取り消す文字が多くて、1文字ずつ削除するのがめんどうな場合は、CTRL＋X（コントロールキーCTRLを押しながらX押す）をキー入力することにより、カーソルが存在する行の全体を取り消すことができます。なお、MS-DOS のバージョン 3.x では、CTRL＋U でも同じことが実現できます。

その実行例を示しましょう。**図 2.13** は、「A＞TYPE　README.DOC」というコマンドをキー入力したものの、それを実行する前に DIR コマンドを実行したくなり、入力した行全体を CTRL＋X で取り消して、新たに DIR コマンドをキー入力した例です。

```
A>TYPE README.DOC¥
  DIR

 ドライブ A: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは A:¥

COMMAND  COM     24931  88-07-13   0:00
FORMAT   EXE     97766  88-07-13   0:00
MP       COM     36210  87-07-12   3:10
MP       COD     71904  87-07-12   3:10
MP       SYS    132656  87-07-12   3:10
MP       DAT     29162  87-07-12   3:10
MP       HLP    102194  87-07-12   3:10
ATOK6A   SYS     54271  87-06-15  12:00
ATOK6B   SYS     21372  87-06-15  12:00
ATOK     DIC    454144  89-07-19  20:10
CONFIG   SYS        78  89-07-19  15:12
AUTOEXEC BAT        27  89-07-19  15:12
MP       INI       258  89-07-20  19:16
        24 個のファイルがあります.
     34816 バイトが使用可能です.

A>
```

CTRL＋X を入力する

この行が取り消されたことを示す「¥」が表示され、カーソルが次の行に移る

DIRコマンドを入力し直して実行する

MS-DOSバージョン3.xでは、CTRL＋Uでも1行を取り消すことができる。なおその場合、取り消された行は画面から消える

図 2.13　入力したコマンドの1行全体を取り消す例

このようにCTRL＋X を入力すると、MS-DOS は文字列の最後に「¥」記号を表示してカーソルが次の行に移ります。そのあとに正しいコマンドを入力してください。「¥」の記号は、この記号の付いている行は全部キャンセルされた、という意味を表します。なお、CTRL＋U の場合は、キャンセルされた行は、画面から消えます。

さて、これだけ知っていればこれからの実習は大丈夫でしょう。次に進みます。

2.4 ディスクの内容を見る

前節の図 2.10、や図 2.11 に示されている DIR コマンドによる表示が、そのときのメインのドライブ(ここでは、ドライブＡ：)にセットされているディスクの内容です。これらの表示には、ファイル名のほかにも、ファイルに関するいくつかの情報が含まれています。

```
A>DIR

 ドライブ A: のディスクのボリュームラベルは PC98_SYSTEM
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13    0:00
ADDDRV   EXE    18384  88-07-13    0:00
ASSIGN   EXE    26450  88-07-13    0:00
ATTRIB   EXE     9154  88-07-13    0:00
APPEND   COM     2052  88-07-13    0:00
BACKUP   EXE    25218  88-07-13    0:00
CHKDSK   EXE    10384  88-07-13    0:00
CUSTOM   EXE    34566  88-07-13    0:00
DELDRV   EXE     9842  88-07-13    0:00
DISKCOPY EXE    10604  88-07-13    0:00
MP       INI     5039  88-07-13    0:00
OFG      INI     5494  88-07-13    0:00
TELE     INI     3855  88-07-13    0:00
MENU     1          7  88-07-13    0:00
MENU     2         13  88-07-13    0:00
README   DOC    10578  88-07-13    0:00
      57 個のファイルがあります.
   26624 バイトが使用可能です.

A>
```

図 2.14　DIR コマンドの実行例

では、表示されているそれぞれの部分の意味を解説していきましょう。

図 2.15 に示すのは、**ボリュームラベル**が表示される部分です。ここでは、Ａ：のディスクドライブにセットされているディスクには、「PC98_SYSTEM」という名前(ボリュームラベル)が付いていることを表しています(オリジナルのシステムディスクには、図 2.10 に示したようにボリュームラベルは付いていない。これはそのコピーディスクであり、ボリュームラベルを付けてある。ボリュームラベルの付け方などは 7.14 節で解説する)。

```
A>DIR

 ドライブ A: のディスクのボリュームラベルは PC98_SYSTEM
 ディレクトリは A:¥

COMMAND  COM     24931  88-07-13    0:00
ADDDRV   EXE     18384  88-07-13    0:00
ASSIGN   EXE     26450  88-07-13    0:00
  ～～～～～～～～～～～～～～～～～～～～～～～
MENU     2          13  88-07-13    0:00
README   DOC     10578  88-07-13    0:00
        57 個のファイルがあります.
     26624 バイトが使用可能です.

A>
```


図2.15　DIRコマンドの表示内容　「ボリュームラベル」の部分に注目

図2.16に示すのは、**ディレクトリ**が表示される部分で、ここには、現在表示されている**ファイル**が収容されているディスク上の区域名(**ディレクトリ名**)が表示されます。この例の表示「A：¥」は、現在表示されているファイルが収容されている区域が、ディスクドライブA：のディスクの、いちばんもとの区域(「¥」の区域を**ルートディレクトリ**と呼ぶ)であることを意味しています。

```
A>DIR

 ドライブ A: のディスクのボリュームラベルは PC98_SYSTEM
 ディレクトリは A:¥

COMMAND  COM     24931  88-07-13    0:00
ADDDRV   EXE     18384  88-07-13    0:00
ASSIGN   EXE     26450  88-07-13    0:00
  ～～～～～～～～～～～～～～～～～～～～～～～
MENU     2          13  88-07-13    0:00
README   DOC     10578  88-07-13    0:00
        57 個のファイルがあります.
     26624 バイトが使用可能です.

A>
```


図2.16　DIRコマンドの表示内容　「ディレクトリ」の部分に注目

このように、画面のちょっとしたメッセージにも、私たちがまだ学んでいないことがたくさん出てきますが、これらはおいおい学んでいきますので、今の時点では問題にしないでけっこうです。

さて、これらの表示の下にズラズラと並んでいるのが、その区域に存在するファイルの一覧表です。メッセージの最後に表示されているように、この区域には全部で57個のファイルがあります。

図2.17の中で、最後のファイル「README.DOC」に注目してみましょう。DIRコマンドの実行によって各ファイルのファイル名、ファイルの大きさ、ファイルの作成(更新)日時が、ファイルごとに表示されています。

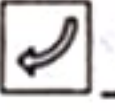
```
A>DIR

 ドライブ A: のディスクのボリュームラベルは PC98_SYSTEM
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13   0:00
ADDDRV   EXE    18384  88-07-13   0:00
ASSIGN   EXE    26450  88-07-13   0:00
ATTRIB   EXE     9154  88-07-13   0:00
APPEND   COM     2052  88-07-13   0:00
BACKUP   EXE    25218  88-07-13   0:00
CHKDSK   EXE    10384  88-07-13   0:00
         BAT       59  88-07-13
CONFIG   SYS       88  88-07-13   0:00
MENU     MNU     5051  88-07-13   0:00
SAMPLE   MNU     5402  88-07-13   0:00
SETUP    DEF     1941  88-07-13   0:00
MP       INI     5039  88-07-13   0:00
OFG      INI     5494  88-07-13   0:00

README   DOC    10578  88-07-13   0:00
    57 個のファイルがあります.
 26624 バイトが使用可能です.

A>
```


図2.17　DIRコマンドの表示内容　個々のファイルに関する情報に注目

この例では、57個のファイルがDIRコマンドで表示されています。これがドライブA：のディスクのルートディレクトリ(いちばんもとの区域)に収容されている内容のすべてです。厳密には、起動する前のMS-DOS自身もこのディスク上に**ファイル**として記録されているのですが、これはDIRコマンドでは表示されません。

私たちが、MS-DOS上でディスクに関して何かを行う場合、その対象となるのは、すべてこのような、DIRコマンドで表示されるファイルです(ファイル名やファイルの種類など、ファイルに関して詳しくは6章で解説する)。

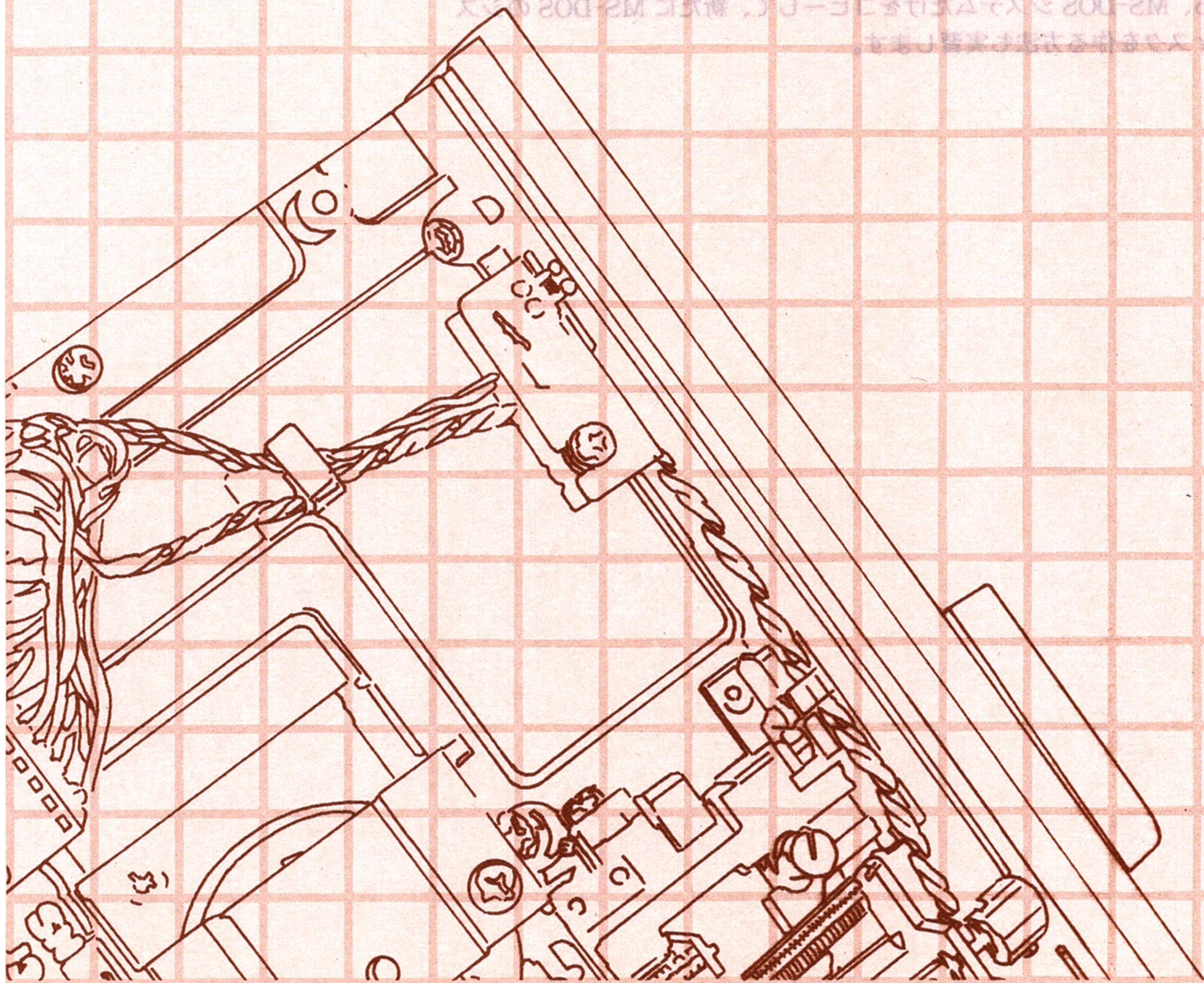
また、DIRコマンドには、次のような表示形式のバリエーションがあります。「DIR」の後ろに、続けて「/W⏎」とキー入力してください。その実行例を示しましょう(図2.18)。「DIR/W」あるいは、「DIR　/W⏎」とキー入力します(つまり、「/」の前にスペースがあってもなくてもよい)。

```
A>DIR/W ⏎ ……………DIRコマンドのワイド表示を実行する

 ドライブ A: のディスクのボリュームラベルは PC98_SYSTEM
 ディレクトリは A:¥

COMMAND  COM     ADDDRV   EXE    ASSIGN   EXE    ATTRIB   EXE    APPEND   COM
BACKUP   EXE     CHKDSK   EXE    CUSTOM   EXE    DELDRV   EXE    DISKCOPY EXE
DUMP     EXE     EDLIN    EXE    FC       EXE    FORMAT   EXE    KEY      EXE
MENU     COM     MENUED   EXE    MSASSIGN COM    RENDIR   COM    REPLACE  EXE
RESTORE  EXE     SPEED    EXE    SWITCH   EXE    SYMDEB   EXE    SYS      EXE
TREE     EXE     USKCGM   EXE    XCOPY    EXE    EMSDRIVE SYS    GRAPH    LIB
GRAPH    SYS     MOUSE    SYS    NECAIK1  DRV    NECAIK2  DRV    PRINT    SYS
RAMDISK  SYS     RSDRV    SYS    BATKEY   COM    CHKFIL   EXE    CHKENV   COM
SETUP    EXE     SETUP2   EXE    AUTOEXEC BAT    HDCOPY   BAT    HDFORM   BAT
MPX      BAT     OFGX     BAT    CONFIG   SYS    MENU     MNU    SAMPLE   MNU
SETUP    DEF     MP       INI    OFG      INI    TELE     INI    MENU     1
MENU     2       README   DOC
        57 個のファイルがあります.
     26624 バイトが使用可能です.

A>
```

図2.18　DIRコマンドのワイド表示例

このように、1行に5個のファイル名が表示され、1画面にたくさんのファイル名が表示されました。その代わり、ほかの情報は省略されています。

さて本章では、MS-DOSを起動し、最も多用するDIRコマンドを使って、ディスクの内容を見るところまで話が進みました。次章では、ディスクのバックアップコピーを作る実習をしましょう。

3章
ディスクのバックアップコピーを作る

本章では、ディスクの最も基本的な操作である「フォーマット処理」「ディスクのコピー」および「ファイルのコピー」について解説します。

新しいディスクを使うには、フロッピーディスク、ハードディスク、いずれの場合にも、最初に１度だけは「フォーマット処理」を行う必要があります。また、ディスクの内容をコピーするには、いくつかの方法がありますが、それらの操作には、いずれの場合にも「ファイル」というものの知識が必要です。本書を読み進み、その辺のことが理解できるようになれば、ビジネスソフトのユーザーにとっても、日常の仕事に大いに役立てることができると同時に、コンピュータのいろいろなことに自信が持てるようになるでしょう。

今後の実習には、自由に書き込みや消去をしてもかまわない（内容を壊しても差し支えない）システムディスクや、ビジネスソフトのディスクが必要となりますので、まず最初に、そのためのコピーディスクを作る実習をしましょう。

また、日常使っている、MS-DOS の組み込まれたビジネスソフトのディスクから、MS-DOS システムだけをコピーして、新たに MS-DOS のシステムディスクを作る方法も実習します。

3.1 使用するフロッピーディスクのフォーマット処理

ディスクは、フロッピーディスクでもハードディスクでも、購入したままの状態では使用することができません。これはMS-DOSに限らず、すべてのOSについて同じです。

あらゆるディスク装置のすべてのメディア(記録媒体)は、最初に1度だけ、**フォーマット処理**と呼ばれる前処理を必ず行わなければなりません(この前処理は、**イニシャライズ**とか、**初期化**などとも呼ばれる)。

ここではフロッピーディスクに関して話を進めましょう。まず、フロッピーディスクの外袋(ジャケット)の中で回転している磁性体の円盤シートは、毎分300〜400回転程度の、ディスク装置の種類によって決まっている回転数で回転しています。書き込まれるデータは、この円盤シートに同心円状に磁気的に記録されますが、その記録される「道」のことを**トラック**と呼びます。このトラックの本数は、フロッピーディスクの種類によって異なりますが、40本から百数十本程度あります(たとえば、5インチあるいは3.5インチ2HDタイプの場合、裏/表を合わせて154本)。

データが書き込まれているフロッピーディスクの磁性面の様子を、特殊な方法で見たものを**写真3.1**に示します。

撮影協力/富士写真フイルム(株)

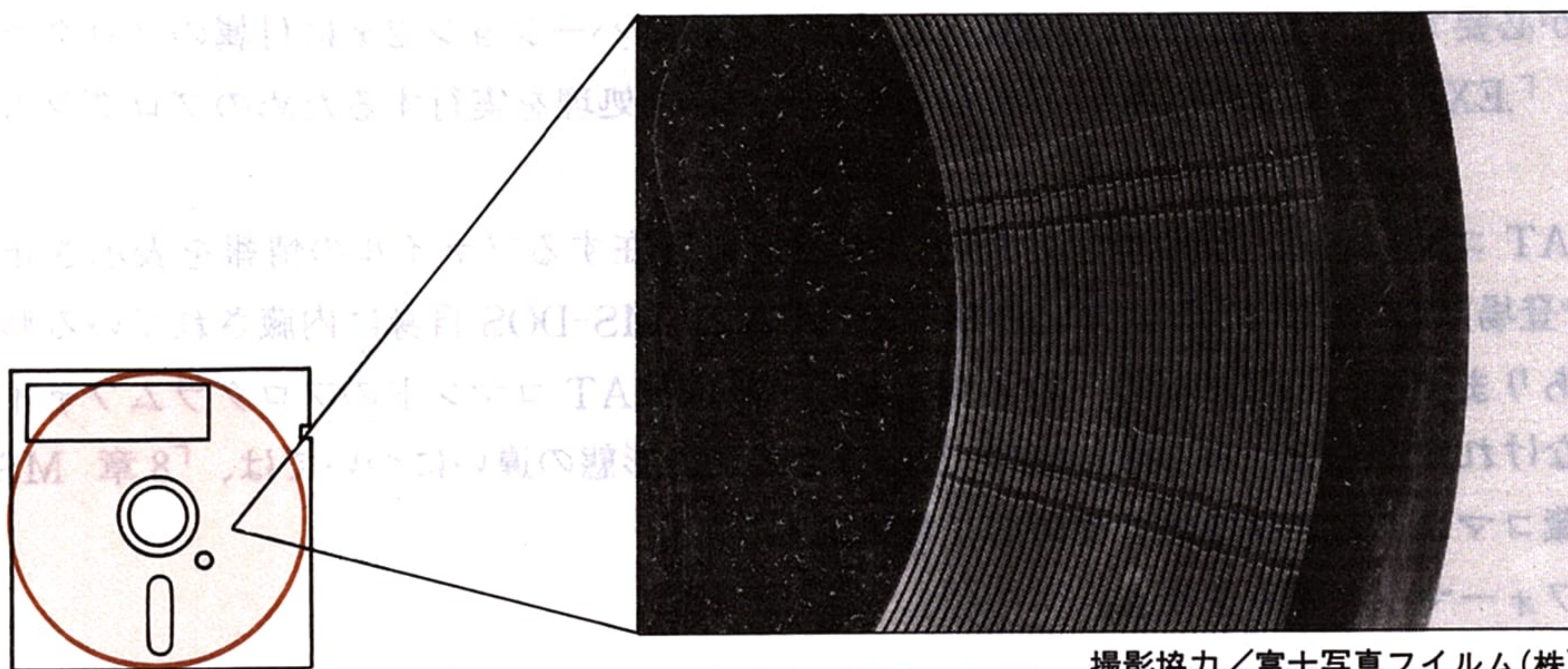



写真3.1　フォーマット処理されたフロッピーディスクのトラックを特殊な方法で見る

新品のディスクを使う前に、必ず1度はフォーマット処理を行うということの意味は、MS-DOSのディスクの記録形式に合わせた「道」を作る(書き込んでおく)ことなのです。「道」の本数やその内部の構造などは、DISK BASICや、CP/Mなどの、それぞれのOSによって異なり、ほかのOSの形式でフォーマット処理されたディスクを、MS-DOSが直接読み出し/書き込みすることはできません。ただし、MSX-DOSやMS-OS/2は、フォーマットの形式が同じですので互いのディスクを読み/書きすることができます。つまりフロッピーディスクの互換性があるわけです。

この「道」――「トラック」には、ディスク装置がディスク上のデータを正確に読み出し／書き込みするために必要な、各種の情報（いわば、いろいろな「標識」）が書き込まれています。ディスクドライブは、そのような各種の情報を持ったトラックが書き込まれていないディスクでは、読み出し／書き込みが不可能なのです。

このようなフォーマット処理は、最初に1度だけ行えばよく、そのあとは通常の読み／書き作業を行っている限り、いったん書き込まれたトラックの情報が消えてしまうことはありません。もっともこのフォーマット処理は、使い古しのディスクの内容を、きれいさっぱり消してしまう目的で、しばしば行うことになるでしょう。

メーカーによっては、自社ブランドのフロッピーディスクを、自社のパソコンのDISK BASICのフォーマットに合わせてフォーマット処理し、「フォーマット済、即使用可能」として発売しているところもあります。しかし、そのようなディスクもMS-DOSで使用するには、新たにMS-DOS上でフォーマット処理をし直す必要があります。購入したディスクがどのような状態にあろうと、新たにMS-DOS上でフォーマット処理を行うことにより、どこのメーカーのどのようなフロッピーディスクでも使用することができるわけです。

ではフォーマット処理を行うための**FORMAT**コマンドを実習してみましょう。このコマンドを実行するには、ディスク上に「FORMAT.EXE」（あるいはFORMAT.COM）という名前のプログラムファイルが必要です。「FORMAT.COM」は、MS-DOSのバージョン2.*x*に付属のプログラムファイルですが、「.EXE」も「.COM」も、どちらもフォーマット処理を実行するためのプログラムファイルです。

FORMATコマンドは、**DIR**コマンド（ディスク上に存在するファイルの情報を表示させるコマンド。2章で登場）のような、そのコマンドのプログラムが、MS-DOS自身に内蔵されている形態のコマンドではありません。したがって、ディスク上に「FORMATコマンドのプログラムファイル」が存在していなければならないのです（コマンドのプログラム形態の違いについては、「8章　MS-DOSの構成と内蔵コマンドおよび外部コマンド」で解説する）。

さて、フォーマットプログラム、

FORMAT.EXE（あるいは .COM）

は、図2.11、2.13のDIRコマンドによる表示の中にも見られるように、MS-DOSのシステムディスクには必ず含まれています。また、ビジネスソフトの常用ディスクなどにも、たいていはコピーされているでしょう。DIRコマンドを実行して確認しておいてください。

FORMAT.EXE（.COM）というファイルが、ディスク上にあることが確認できましたか？　では、ディスクをフォーマット処理する作業を実際に行ってみましょう。なおフロッピーディスクドライブが1台のみのシステムを使っている方も、最初の「ディスクドライブが2台以上の場合」からお読みください。

■ ディスクドライブが2台以上のシステムの場合

MS-DOSを起動して、コマンドレベルの状態(プロンプト「A>」が表示されている状態)にあるとしましょう。まずフォーマット処理されるフロッピーディスクをドライブB：にセットします。このディスクは、新品でも、使い古しでもかまいませんが、中身のデータは、フォーマット処理されることによって、跡形もなく消去されてしまいますので、中身が消されてもよいディスクかどうか、くれぐれも注意してセットしてください。

FORMATコマンドは、次のようにキー入力することにより実行されます。

```
A>FORMAT B:⏎
```

この「B：」の意味は、ドライブB：上のディスクをフォーマット処理する、という意味であり、処理対象のディスクが「ドライブC：」であれば、「C：」とします。もしこの「B：」などのドライブの指定を付け忘れた場合、MS-DOSのバージョン2.xに付属のフォーマットプログラムでは、プロンプトが「A>」であればドライブA：上のディスクがフォーマット処理の対象となりますので注意が必要です。ただし、MS-DOSのバージョン3.xに付属のフォーマットプログラムでは、そのような場合、フォーマットするドライブ名を問い合わせてくるなどの対話形式になっています。

また、ディスクが3台以上接続されているシステムでは、B：に限らず、C：、D：などのドライブを指定することもできますが、「フォーマット処理はいつもドライブB：で」というように、フォーマット処理を行うドライブを習慣として決めておいた方が間違いが少なくてよいでしょう(ただし、ドライブのタイプ(8インチ、5インチ、3.5インチなど)が混在している場合は、そうもいかないが)。

いったん起動したFORMATコマンドの実行を取り止める場合

FORMATコマンドが起動し、何らかのキー入力待ちになっている状態の時、その実行を中止してMS-DOSのコマンドレベルに下りるには、[CTRL]＋[C](コントロールキー[CTRL]を押しながら、[C]を押す)をキー入力します。[CTRL]＋[C]を入力すると同時に、プロンプト「A>」の状態に戻ります。ただしその操作の前に、まずは安全を第一に考え、フォーマット処理しようとしたディスクを取り出しておきましょう。

なお、実際のフォーマット処理が始まってしまってからは(つまりディスクへの書き込みが開始されてからは)、中止することができませんので注意してください。処理が始まってしまえば、たとえ途中で強制的にディスクを取り出しても、そのディスク上のすべてのファイル(それまで使っていたディスクの場合)は、もはや読み／書き不能です。

では、**図3.1**に、これら一連の実行例を示しましょう。

```
A>FORMAT B:⏎ ……FORMATコマンドを起動して、ドライブB:のディスクをフォーマット処理する
Format Version 4.10

新しいディスクをドライブ B: に挿入し ……フォーマット処理されるディスクを正しくセットする
どれかキーを押してください^C ……この時点でこのようにCTRL+Cを入力すると、FORMATコマンドを
                                  中止して、MS-DOSのコマンドレベルに戻ることができる

A>FORMAT B:⏎ ……CTRL+Cの入力によりコマンドレベルに戻ってしまったので、
Format Version 4.10   再度FORMATコマンドを実行する

新しいディスクをドライブ B: に挿入し
どれかキーを押してください x ……適当なキーを入力する

ディスクのタイプは  1 : 640(KB)  2 : 1(MB)  = 2⏎ ……5インチ2HD(1Mバイト)のディスクをフォー
                                                    マット処理するので「2」を入力する
目的のディスクは 1MB FD です

  この間にフォーマット処理が行われる

フォーマットが終了しました
0  10  20  30  40  50  60  70  80  90  100
██████████████████████████████████████████ (%)

  1250304 バイト  全ディスク容量
  1250304 バイト  使用可能ディスク容量

別のディスクをフォーマットしますか(Y/N) N⏎ ……続けて別のディスクをフォーマット処理
A> ……FORMATコマンドが終了して、                 する場合は「Y」を入力する
      コマンドレベルに戻っている

（表示されるメッセージなどは、MS-DOSのバージョンや機種によって異なる）
```

図3.1　フロッピーディスクのフォーマット処理実行例

以上でドライブB：にセットしたフロッピーディスクがフォーマット処理され、MS-DOSで使用できるようになりました。もし何枚かのディスクをフォーマット処理したい場合は、最後の「(Y/N)」の問い合わせに「Y」をキー入力することにより、新しくセットしたディスクを、続けて何枚でもフォーマット処理することができます。

FORMATコマンドには、このほかに

```
A>FORMAT B:/S⏎
```

```
A>FORMAT B:/V⏎
```

```
A>FORMAT B:/S/V⏎
```

など、いろいろいな使い方や機能がありますが、それらについては、3.3 節、および 7.14 節で解説しますので、ここでは FORMAT コマンドの基本的な使い方を学んでください。なお、FORMAT コマンドを起動したあと、実際にフォーマット処理を開始する前に(「どれかキーを押してください」のキーを入力する前に)、念のために対象となるディスク以外の、ほかのディスクがセットしてあるドライブのフタを開けておくことをお勧めします。誤操作による悲劇を避けることができるでしょう。

■ ディスクドライブが 1 台のみのシステムの場合

ディスクドライブが 1 台のシステムでフォーマット処理を行う場合は、FORMAT コマンドを起動するためのドライブと、フォーマット処理するためのドライブが同じということになります。その場合の実行例を示しましょう(図 3.2)。

```
A>FORMAT A:↵ ……………FORMATコマンドを実行する
Format Version 4.10

新しいディスクをドライブ A: に挿入し
どれかキーを押してください x ……………この時点で、フォーマット処理されるディスクと入れ換えて、
                                          適当なキーを入力する

ディスクのタイプは   1 : 640(KB)  2 : 1(MB)  = 2↵ ………5インチ2HD(1Mバイト)のディスクを
                                                  フォーマット処理するので「2」を入力する
目的のディスクは 1MB FD です

    ∫ この間にフォーマット処理が行われている

フォーマットが終了しました
0  10  20  30  40  50  60  70  80  90  100
██████████████████████████████████████████ (%)

    1250304 バイト  全ディスク容量
    1250304 バイト  使用可能ディスク容量

別のディスクをフォーマットしますか(Y/N) N↵ ……………「N」を入力した
A> ……………FORMATコマンドが終了して、
             コマンドレベルに戻っている
           (表示されるメッセージなどは、MS-DOSのバージョンや機種によって異なる)
```

図 3.2　フロッピーディスクドライブ 1 台のみのシステムにおけるフォーマット処理実行例

この場合は、ディスクの入れ換えを間違えないように十分注意し、オリジナルディスクをフォーマット処理してしまわないよう、慎重に作業を行ってください(2.1 節の「書き込み禁止」の項も参照のこと)。

以上でフォーマット処理が終わり、MS-DOS で使用できるディスクが用意できました。次節に進みましょう。

3.2 ディスクのまるごとコピー

「ディスクのまるごとコピー」というのは、ディスク上のファイルに関係なく —— つまり、どのようなファイルがどれだけ存在していようと、また、まったくの空であろうと、ディスクの内容にはいっさいおかまいなく —— 文字どおり機械的に「ディスク全体をそっくりコピー」するものです。このコピー方法は、ファイルをまったく意識せず、ただ単純に、一方のディスクのあるトラックのデータを読み出し、他方のディスクの同じトラックにそのデータを書き写していくもので、ディスクの外周のトラックからコピーを開始し、内側に向って順に全トラックをコピーしていきます。要するにディスク全面の「まるごとコピー」というわけです。

ただし、5インチや3.5インチの2HDのディスク(いずれも容量は1Mバイトのタイプ)と、5インチや3.5インチの2DDのディスク(640Kバイトのタイプ)との間など、ディスクの容量が異なるものの間、あるいは容量が同じでも、トラックの数やセクタ*の数などが異なるディスクの間では、この方法によるコピーはできません。逆に、ディスクの外形が異なっていても、記録容量や、トラック、セクタの数などが同じであればコピーすることができます。これらの関係を**表3.1**に示します。

メディアの種類		3.5インチ		5インチ			8インチ
		2DD	2HD	2D	2DD	2HD	2D
3.5インチ	2DD	○	×	×	○	×	×
	2HD	×	○	×	×	○	○
5インチ	2D	×	×	○	×	×	×
	2DD	○	×	×	○	×	×
	2HD	×	○	×	×	○	○
8インチ	2D	×	○	×	×	○	○

○印はまるごとコピー可
×印はまるごとコピー不可

表3.1　ディスクまるごとコピーが可能な各種フロッピーディスクの相互関係一覧表

---

* ディスク上に記録されるデータの最小単位。5インチや3.5インチ2HDのディスクでは1024バイト。

「ディスクまるごとコピー」を行うには**DISKCOPY**コマンドを使います。このコマンドを実行するには、ディスク上に「DISKCOPY.EXE(あるいは .COM)」という名前のプログラムファイルが必要です。「DISKCOPY.COM」は、MS-DOSのバージョン2.*x*に付属しているプログラムファイルですが、「.EXE」も「.COM」も、どちらも「ディスクまるごとコピー」を実行するためのプログラムファイルです。

このプログラム、

DISKCOPY.EXE(あるいは.COM)

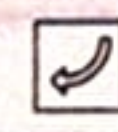
は、図2.14、図2.18のDIRコマンドによる表示の中にも見られるように、MS-DOSのシステムディスクには必ず含まれています。また、日常使っているビジネスソフトなどの常用ディスクにも入れておく場合が多いので、DIRコマンドを実行して確認しておいてください。

「DISKCOPY.EXE(.COM)」というファイルが、ディスク上にあることが確認できましたか？ では2ドライブ以上のシステムでの実行例から示しましょう。ドライブ1台のみのユーザーも、最初の「ディスクドライブが2台以上のシステムの場合」からお読みください。(なお、この「ディスクまるごとコピー」と根本的に異なるコピーに、「ファイル単位のコピー」がある。それについては3.4節で実習解説する)

■ ディスクドライブが2台以上のシステムの場合

ドライブA：にシステムディスクをセットしてMS-DOSを起動し、ドライブB：にフォーマット済みのディスク(MS-DOSで使用していた、いろいろなファイルが書き込まれているディスクでもかまわない)をセットします。次にDISKCOPYコマンドを次のようにキー入力して実行します。

A>DISKCOPY　A：　B：⏎

この「A：　B：」の意味は、ドライブA：からドライブB：へコピーするというデータの流れの方向を指示しています。つまり、

A：　→　B：
A：　to　B：
A：　から　B：　へ

という意味であり、MS-DOSのコマンドは、このDISKCOPYコマンドに限らず、方向づけが必要なコマンドのすべては、左側から右側に物事が流れるように書きます。

たとえばDISKCOPYコマンドの場合、上の例と逆にして、

A>DISKCOPY　B：　A：⏎

とすると、ドライブB：のディスクの内容がドライブA：のディスクにそっくりコピーされてしまいます。このように逆に書いてしまうと、結果的にドライブA：にセットしたMS-DOSのシステムディスクや、ビジネスソフトのディスクが、ドライブB：のディスクと同じ内容に書き換えられてしまいますので、コピーの方向を示す書き順を絶対に間違えないよう、ディスプレイに表示されるメッセージにも十分注意して実行してください。

いったん起動したDISKCOPYコマンドの実行を取り止める場合

DISKCOPYコマンドが起動し、何らかのキー入力待ちになっている状態のとき、実行を中止してMS-DOSのコマンドレベルに下りる場合は、CTRL＋Cをキー入力します。CTRL＋Cを入力すると同時に、プロンプト「A＞」の状態に戻ります(3.1節のFORMATコマンドの実行を取り止める場合と同じ)。では、これら一連の実行例を示しましょう(図3.3)。

```
A>DISKCOPY A: B:⏎ ……………DISKCOPYコマンドを実行する。この場合は、ディスクA:をディスクB:へコピーする

DISKCOPY version 4.00

ディスクのコピーを行います

送り側ディスクをドライブ A: に挿入してください
受け側ディスクをドライブ B: に挿入してください
準備ができたらどれかのキーを押してください^C ……この時点でこのようにCTRL+Cを入力すると、
                                               DISKCOPYコマンドを中止してコマンドレベル
                                               に戻ることができる
A>DISKCOPY A: B:⏎ ……………CTRL+Cの入力により、コマンドレベルに戻ってしまった
                           ので、再度DISKCOPYコマンドを実行する
DISKCOPY version 4.00

ディスクのコピーを行います

送り側ディスクをドライブ A: に挿入してください } このメッセージに従って、それぞれの
受け側ディスクをドライブ B: に挿入してください } ディスクを正しくセットする
準備ができたらどれかのキーを押してくださいx ……………適当なキーを入力する

    ∫ この間にコピー作業が行われている

コピーは終了しました
もう一度実行しますか(Y/N)N ……………続けて別のディスクをコピーする場合は「Y」を入力する


A> ……………DISKCOPYコマンドが終了して、コマンドレベルに戻っている

                    (メッセージの表示などは、MS-DOSのバージョンや機種によって異なる)
```

図3.3　ディスクのまるごとコピー実行例

以上でドライブA：のディスクの全内容が、そっくりドライブB：側のディスクにコピーされました。以降の実習には、ここで作ったコピーディスクを使うことにして、オリジナルディスクは大切に保管しておきましょう。

もし、同じコピーディスクを何枚か作りたい場合や、別のオリジナルディスクをコピーしたい場合は、最後の「(Y/N)」の問い合わせに「Y」をキー入力して、B：側やA：側のディスクを入れ換えれば、続けて何度でもDISKCOPYコマンドを実行することができます。

ディスクが3台とか4台とか接続されているシステムでは、

```
A>DISKCOPY A: C:⏎
```

```
A>DISKCOPY B: D:⏎
```

など、いろいろいな使い方をすることができますが、これも前節のFORMATコマンドのときに述べたように、習慣として、たとえば「ディスクコピーはA：からB：へ」というように決めておいた方が間違いが少ないでしょう。

また、このDISKCOPYコマンドには、付加機能として、ディスクのコピーは行わずに、2つのディスクの内容を比較して、一致しているか、相違があるかをチェックする機能が付いているものもあります。これについては、7.11節で実習していますので参照してください。

ドライブが2台以上のコンピュータシステムをお使いの方は、次の項はパスして、3.3節へジャンプしてもかまいません。

■ ディスクドライブが1台のみのシステムの場合

ディスクドライブが1台のシステムで「ディスクまるごとコピー」を行う場合には、かなりめんどうな操作になります。表示されるメッセージに従って、オリジナルディスクとコピー先のディスクを入れ換えながら、**図3.4**のように実行します。

実際にこの作業を行う場合には、ディスクの入れ換えに十分注意してください。入れ換えを誤ると、オリジナルディスクを壊してしまいます。

```
A>DISKCOPY A: A:↵ ……………DISKCOPYコマンドを実行する

DISKCOPY version 4.00

ディスクのコピーを行います

送り側ディスクをドライブ A: に挿入してください

準備ができたらどれかのキーを押してください x



受け側ディスクをドライブ A: に挿入してください

準備ができたらどれかのキーを押してください x


送り側ディスクをドライブ A: に挿入してください

準備ができたらどれかのキーを押してください x

読み込み中　残り 154 トラックです

    ∫

同じことの繰り返し

    ∫

コピーは終了しました
もう一度実行しますか(Y/N)N


A> ……………DISKCOPYコマンドが終了して、コマンドレベルに戻っている
```

これらのメッセージに従って、間違いのないようにディスクをセットする（xは、適当なキーの入力）。内部的には、送り側のディスクから数トラック分のデータを読み、コンピュータのメモリに読み込み、それを受け側のディスクの同じトラックに書き込む動作を何回も繰り返す。かなりめんどうな作業となる

（メッセージの表示などは、MS-DOSのバージョンや、機種によって異なる）

図 3.4　ディスクドライブが 1 台の場合におけるディスクのまるごとコピー実行例

以上のような作業で、MS-DOS のシステムディスクや、ビジネスソフトなどのコピーディスクができました。これ以降の実習には、ここで作ったコピーディスクを使い、それぞれのオリジナルディスクは大切に保管しておきましょう。

3.3 MS-DOS 部だけのコピー

本節では、マザーとなる MS-DOS システムディスクから、MS-DOS 部のみをコピーして、新しい MS-DOS システムディスクを作成するためのコマンドを実習します。

前節では、ディスクをまるごとコピーする DISKCOPY コマンドを使って、MS-DOS のシステムディスクや、ビジネスソフトなどのバックアップコピーのディスクを作りました。このまるごとコピーの方法は、操作が簡単で、ディスクの全内容を高速にコピーできるため、私たちは今後、このコマンドをしばしば利用することになるでしょう。しかし DISKCOPY コマンドは、ディスク上の MS-DOS の部分だけとか、任意のファイルだけなどを、選択してコピーすることが原理的にできません。必要なものも、不要なものも、また、秘密にしておかなければならないものも、ディスク上のデータは何もかもみんなコピーしてしまいます。

そこで MS-DOS には、さらに 2 つのコピーに関するコマンドが用意されています。前節の DISK-COPY コマンドと合わせて、とりあえず 3 つの形式、合計 4 種類のコピーに関するコマンドがあることになります。それをまとめて**表 3.2** に示します。

コピーの形式	コマンド名
ディスクをまるごとコピー	DISKCOPY
MS-DOS 部だけをコピー （システムのコピー）	/S スイッチ付きの FORMAT コマンド （COMMAND.COM は同時にコピーされる）
	SYS （COMMAND.COM はコピーされない）
任意のファイルをコピー	COPY

表 3.2　コピーに関する 3 つの形式と 4 種類のコマンド

この表の「コマンド名」の欄は、コマンドの主命令部だけですので、それぞれの使い方などは、本文の各項目を参照してください。

■ FORMAT コマンドの/S スイッチによるコピー

では、表 3.2 にある「MS-DOS 部だけをコピー」の実習解説をしましょう。まず最初は、FORMAT コマンドの「/S」スイッチを使う場合です。

3.1 節で、ディスクのフォーマット処理について実習しました。そこでも少し触れましたが、FOR-MAT コマンドには、フォーマット処理を行う本来の機能のほかに、

A＞FORMAT　B：/S ⏎

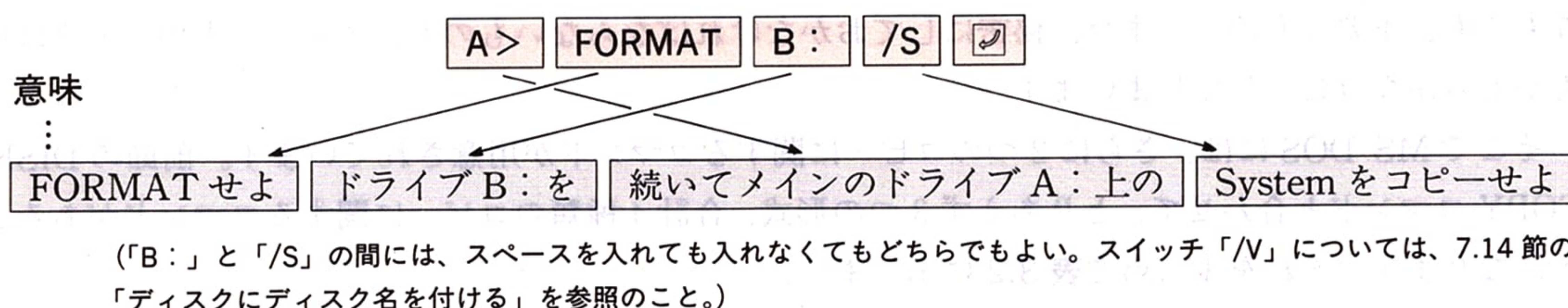
A＞FORMAT　B：/V ⏎

などのコマンドによる付加機能(オプション)があります。この「/S」「/V」などは、FORMAT コマンドの付加機能を働かせるためのもので、**スイッチ、コマンドオプション**とか呼ばれています。/S スイッチの「S」は、System(MS-DOS のシステム)のことであり、このスイッチを付けて FORMAT コマンドを実行すると、フォーマット処理とともに、プロンプトに示されているメインのドライブ(「A＞」ならば A：、「B＞」ならば B：)上の MS-DOS システムがコピーされます。この場合の FORMAT コマンドは、次のように実行します。

A＞ | FORMAT | B： | /S | ⏎

意味
⋮

FORMAT せよ | ドライブ B：を | 続いてメインのドライブ A：上の | System をコピーせよ

(「B：」と「/S」の間には、スペースを入れても入れなくてもどちらでもよい。スイッチ「/V」については、7.14 節の「ディスクにディスク名を付ける」を参照のこと。)

では実行例を示しましょう。MS-DOS を起動して、コマンドレベルの状態(プロンプト「A＞」が表示されている状態)にあるとしましょう。まず、ドライブ B：に内容が失われてもよいディスク(フォーマットされていないディスクでもよい)をセットして、/S スイッチ付きの FORMAT コマンドを実行します(図 3.5)。

```
A>FORMAT B:/S ⏎ ……………/Sスイッチ付きのFORMATコマンドを実行する
Format Version 4.10

新しいディスクをドライブ B: に挿入し……………フォーマットされるディスクを正しくセットする
どれかキーを押してください x ……………適当なキーを入力する

ディスクのタイプは　 1 : 640(KB)　2 : 1(MB)　=　2 ⏎ ……………目的のディスクのタイプを指定する

目的のディスクは 1MB FD です

　　∫ フォーマット処理が行われる

フォーマットが終了しました
```

— 図 3.5 —（次ページに続く）

```
システムを転送しました…………フォーマット作業が終了してからメインのドライブ(A:)の
                              システム部がコピーされる

  1250304 バイト  全ディスク容量…………ディスクの全容量*
   120832 バイト  システム領域 …………ディスク中、MS-DOS部が占める容量(図3.1の実行例には、
                                                                    この領域はない)*
  1129472 バイト  使用可能ディスク容量…………ユーザーが使用できる容量*

別のディスクをフォーマットしますか(Y/N) N⏎…………続ける場合は「Y」を入力する
A> ……………FORMATコマンドが終了して、
           コマンドレベルに戻っている
                              *システム領域などの容量はMS-DOSバージョン3.3の例
```

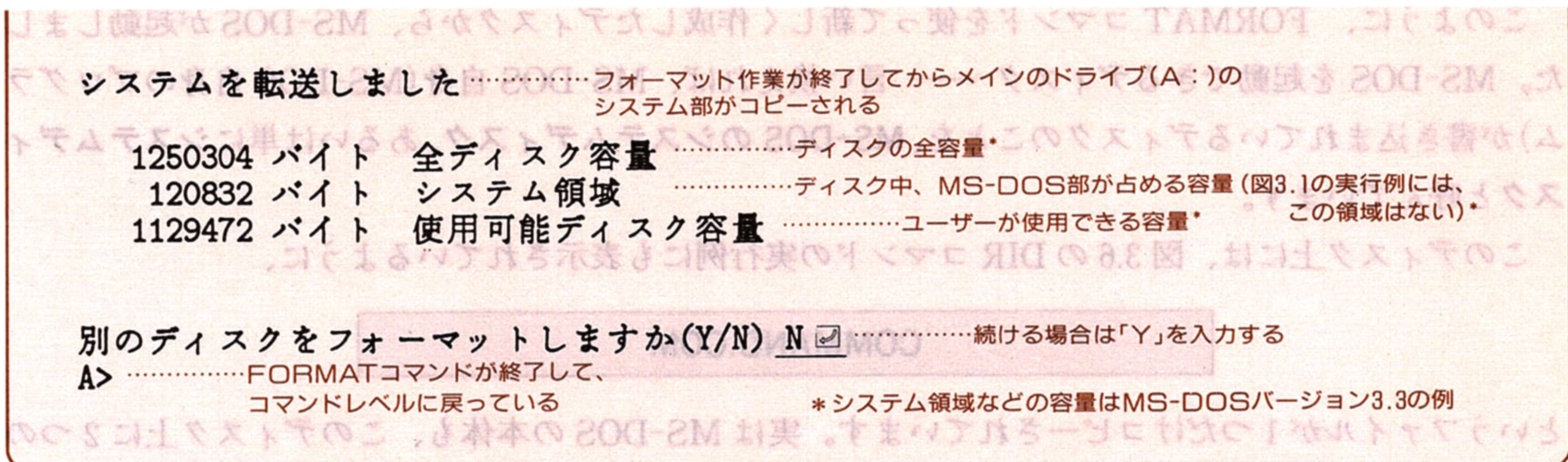



図 3.5　FORMAT コマンドの/S スイッチによる MS-DOS システムのコピー実行例

この結果、ドライブ B：のディスクがフォーマット処理され、そのあと、ドライブ A：のディスク上の MS-DOS システムがコピーされました(プロンプトに示されているメインのドライブ上の MS-DOS システムがコピーされる)。新しい MS-DOS システムディスクができたわけです。

確認のため、そのドライブ B：のディスクを取り出してドライブ A：にセットし、リセットボタンを押して、新しく作成したシステムディスクの MS-DOS を起動してみましょう(図 3.6)。

```
(リセットボタンを押す)
NEC PC-9800 Series Personal Computer
                                                   MS-DOSのオープニングメッセージ。
ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30                          MS-DOSが起動した
Copyright 1981,88 Microsoft Corp. / NEC Corporation

Command ﾊﾞｰｼﾞｮﾝ 3.30
現在の日付は 1989-07-19 (水) です.
日付を入力してください:⏎
現在の時刻は 17:03:48.00 です.  ……日/時を変更しない場合は⏎を入力する
時刻を入力してください:⏎

A>DIR⏎ ……………MS-DOSが起動し、コマンドレベルになったので、DIRコマンドを実行する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13   0:00 …………このようにFORMATコマンドの/Sスイッチ
        1 個のファイルがあります.                によって、MS-DOSシステムと同時にファイ
  1129472 バイトが使用可能です.                  ルCOMMAND.COMがコピーされている

A>
```

図 3.6　FORMAT コマンドの/S スイッチによって作成された MS-DOS のシステムディスクを起動する

このように、 FORMATコマンドを使って新しく作成したディスクから、MS-DOSが起動しました。MS-DOSを起動できるディスク —— 言い換えれば、MS-DOS自身(MS-DOS自身のプログラム)が書き込まれているディスクのことを、**MS-DOSのシステムディスク**、あるいは単に**システムディスク**と呼んでいます。

このディスク上には、図3.6のDIRコマンドの実行例にも表示されているように、

COMMAND.COM

というファイルが1つだけコピーされています。実はMS-DOSの本体も、このディスク上に2つのファイルとして存在しているのですが、それらのファイルはDIRコマンドでは表示されない特別なファイルです。それに対してCOMMAND.COMは、普通のプログラムファイルと同じ形式のファイルですが、MS-DOSシステムの一部分を構成する重要なプログラムであり、**コマンド・プロセッサ**と呼ばれています。このファイルがディスク上にない場合、MS-DOSはプロンプト「A>」を表示して(コマンドレベルの状態で)立ち上がることができません。つまり、MS-DOSを起動できる最も基本的で最小限の(いわば「裸のMS-DOS」)システムディスクは、次のようなものになります。

MS-DOSの本体 (DIRコマンドでは表示されない2つのファイル)	＋	COMMAND.COM (DIRコマンドで表示される1つのファイル)

MS-DOSシステムディスクとは、最低限これら3つのファイルが書き込まれているディスクであるわけです(詳しくは、「4.2　MS-DOSシステムディスク」で解説)。

以上で、/Sスイッチを付けたFORMATコマンドを使って、最も基本的なMS-DOSシステムディスクを作成することができました。

では次に、MS-DOSシステムをコピーするためのもう1つの方法 —— システムコピー専用コマンドの「SYS」を使って、システムディスクを作ってみましょう。

■ システムコピーコマンドSYSによるコピー

各メーカーがそれぞれの機種用に提供しているMS-DOSシステムディスクには、MS-DOSのシステムをコピーする専用コマンドであるSYSコマンドのプログラムファイル、

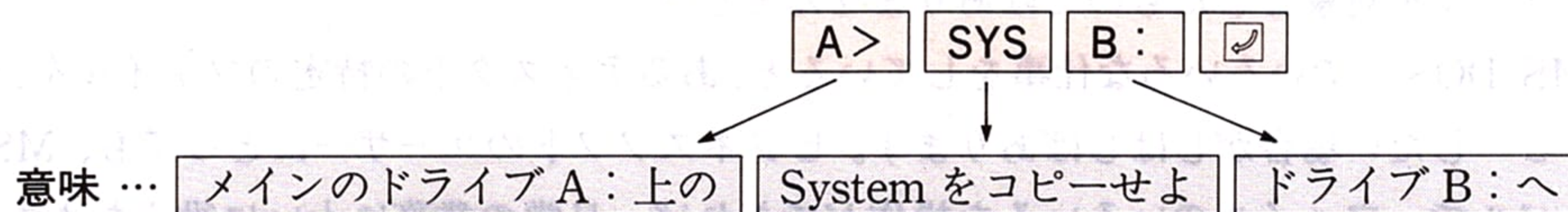
SYS.EXE(あるいは.COM)

が必ず含まれています。SYSコマンドを実行するには、このプログラムファイルが必要ですので、まず、実習に使うシステムディスク上に、このファイルが存在しているかどうかを、DIRコマンドで確認してください。

では、SYS コマンドを使って、MS-DOS システムをコピーしてみましょう。MS-DOS を起動し、プロンプト「A>」が表示されている状態で、ドライブ B：には、/S スイッチを付けずにフォーマット処理したままの、空のディスクをセットします。

SYS コマンドは、次のように実行します。

A> SYS B： ⏎

意味 … メインのドライブ A：上の ／ System をコピーせよ ／ ドライブ B：へ

この SYS コマンドを実行するときのプロンプトが「A>」であれば、「メインのドライブ＝A：」ですので、ドライブ A：の MS-DOS システムがドライブ B：にコピーされます（もしプロンプトが「C>」であれば、ドライブ C：の MS-DOS システムがコピーされる）（**図 3.7**）。（プロンプトの表示と「メインのドライブ」との関係は、7.3 節参照）

```
A>SYS B:⏎ ……………SYSコマンドを実行して、メインのドライブ(A：)のシステム部を
                   ドライブB:のディスクにコピーする

 ＳＹＳコマンド  Version 1.1
Copyright (C) 1988 NEC Corporation

システムが転送されました

A>DIR B:⏎ ……………SYSコマンドが終了してコマンドレベルに戻ったので、DIRコマンドを
                   実行し、ドライブB:上のディスクの内容を確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

ファイルが見つかりません.……………ディスクB:にはDIRコマンドで表示されるファイルはない
                              (COMMAND.COMはコピーされていない)

A>
```

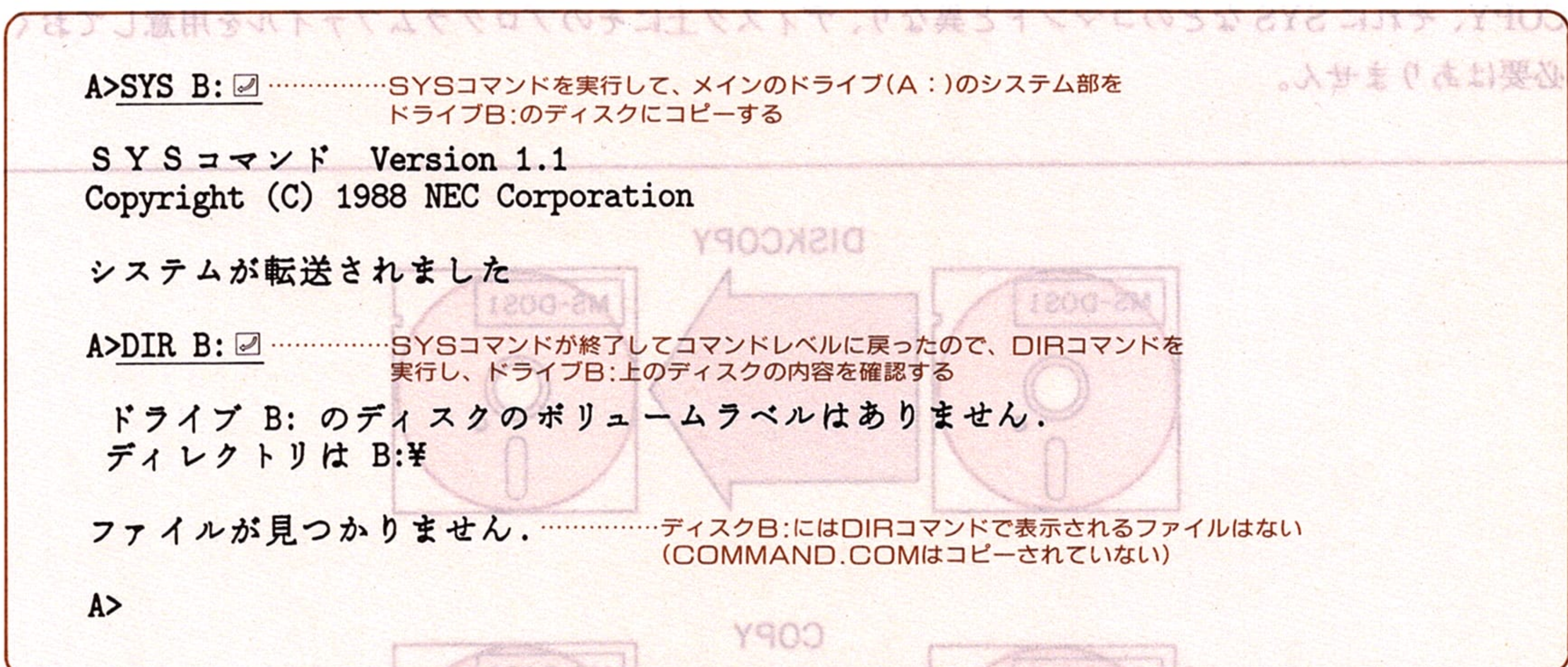



図 3.7　SYS コマンドによる MS-DOS システムのコピー実行例

SYS コマンドの実行が終わり、ドライブ B：のディスクには、ドライブ A：のディスクの MS-DOS システムがコピーされました。しかし、実行例の DIR コマンドの表示でもわかるように、SYS コマンドでは COMMAND.COM ファイルはコピーされません。このファイルがないと、MS-DOS を起動することができませんので、ファイルをコピーするコマンド「COPY」を使って、COMMAND.COM ファイルをドライブ B：にコピーする必要があります（次節で実習する）。SYS コマンドを使う場合にはこの点に注意してください。

3.4 任意のファイルのコピー

これまで実習したコピーコマンドは、ディスク全体のコピーと、MS-DOS システムのコピーであり、個々のファイルを対象としたものではありませんでした。

ところが MS-DOS 上でいろいろな仕事をしていると、あるディスク上の特定のファイルを、任意のディスクにコピーしたい場合がしばしばあります。ビジネスソフトのユーザーにとっても、MS-DOS のコマンドレベルで、ファイルのいろいろな操作ができれば、日常の業務に大いに役立ちます。その効果は絶大であり、ほんの入門的な知識だけでも身に付けておけば、同じアプリケーションソフトを活用するにも大きな差が出るでしょう。

さて、ファイルのコピーには、**COPY** コマンドを使いますが、ここではその基本的な使い方を実習します。COPY コマンドは、DIR コマンドと同じ形態の**内蔵コマンド**であり、これを実行するためのプログラムは、MS-DOS 自身に内蔵されています。ですからこれまでに実習した、FORMAT や、DISK-COPY、それに SYS などのコマンドと異なり、ディスク上にそのプログラムファイルを用意しておく必要はありません。

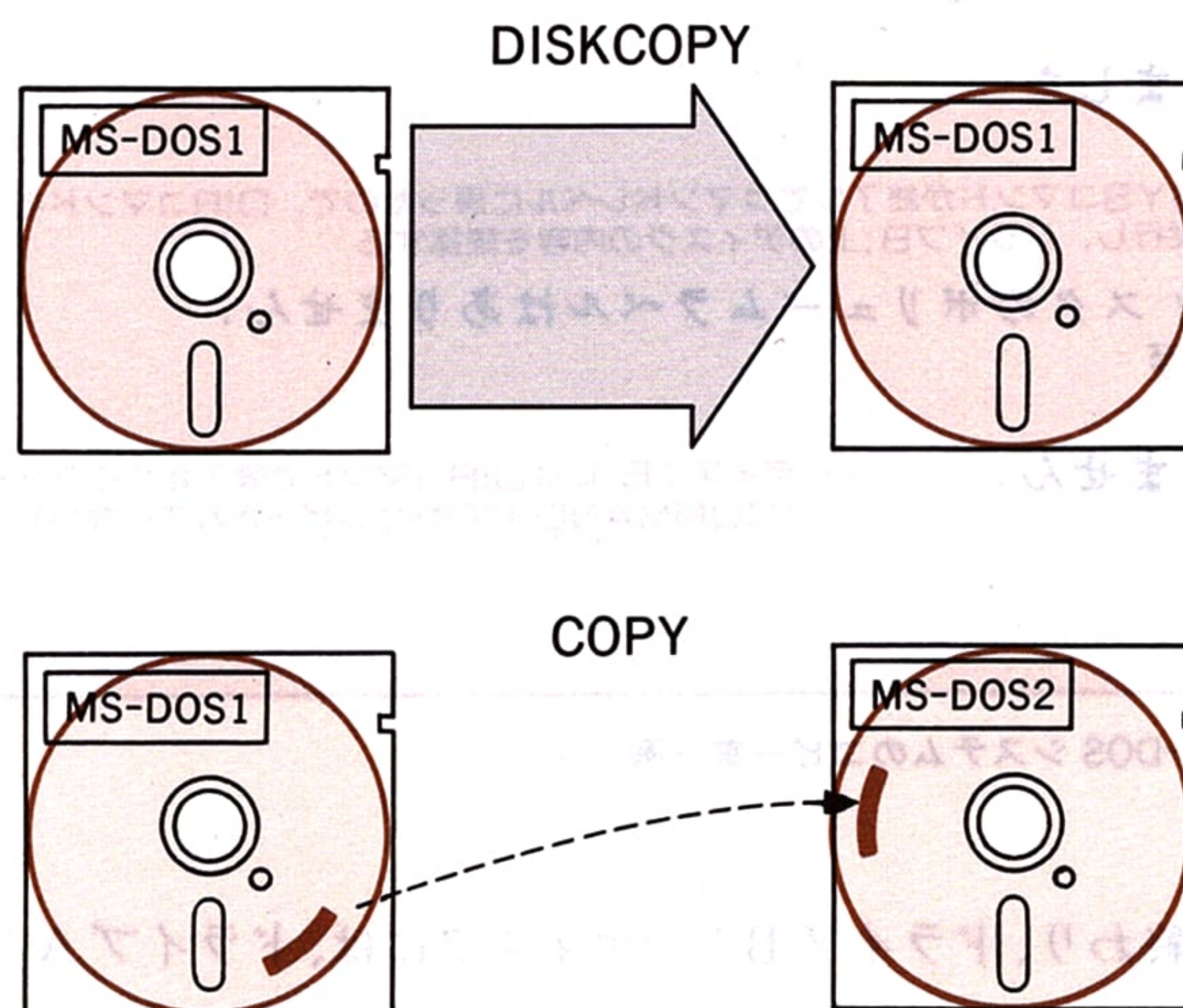
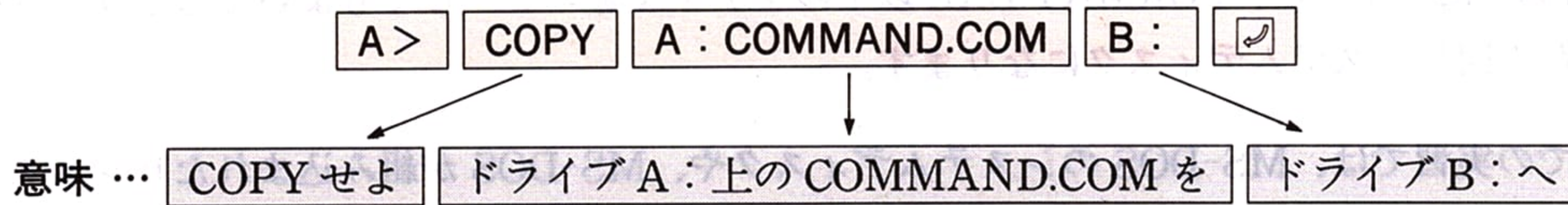



図 3.8　DISKCOPY と COPY コマンドの違い

では、ドライブ A：上の任意のファイルを、ドライブ B：にコピーしてみましょう。ドライブ B：のディスクは、フォーマット処理しただけのディスクでも、システムディスクでも、ほかのファイルがコピーされているディスクでもかまいません。ここでは、前節の SYS コマンドによる作業の続きを

想定して、MS-DOSシステムの本体のみがコピーされたディスクに、ドライブA：上のファイル「COMMAND.COM」をコピーしてみましょう。COPYコマンドは、次のように実行します。

A> | COPY | A：COMMAND.COM | B： | ↵

意味 … COPYせよ | ドライブA：上のCOMMAND.COMを | ドライブB：へ

なお、図3.9に示す実行例では、「COMMAND.COM」のほかに、ついでに「FORMAT.EXE」「DISKCOPY.EXE」「SYS.EXE」の、合計4つのプログラムファイルをコピーしています。

```
A>COPY A:COMMAND.COM B:↵
        1 個のファイルをコピーしました.

A>COPY FORMAT.EXE B:↵
        1 個のファイルをコピーしました.

A>COPY DISKCOPY.EXE B:↵
        1 個のファイルをコピーしました.

A>COPY SYS.EXE B:↵
        1 個のファイルをコピーしました.

A>DIR B:↵

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

COMMAND  COM     24931  88-07-13    0:00
FORMAT   EXE     97766  88-07-13    0:00
DISKCOPY EXE     19604  88-07-13    0:00
SYS      EXE     25480  88-07-13    0:00
         4 個のファイルがあります.
    985088 バイトが使用可能です.

A>
```


図3.9 COPYコマンドによるファイルコピーの実行例

以上でドライブB：のディスクには、前節のSYSコマンドでコピーされた、**MS-DOSシステム**(DIRコマンドでは表示されない2つのファイル)のほかに、今回、図3.9に示されているように、4つのファイル(「COMMAND.COM」「FORMAT.EXE」「DISKCOPY.EXE」「SYS.EXE」)がCOPYコマンドでコピーされ、新しいMS-DOSシステムディスクができ上りました。前回と同様、ここで作成したディスクをドライブA：にセットしてリセットボタンを押し、MS-DOSが起動することを確認しておいてください。今後の実習には、このディスクを利用します。

また、このようなシステムディスクを作成するには、前節で実習したFORMATコマンドの/Sスイッチを使って作成したシステムディスク(COMMAND.COMは自動的にコピーされている)に、図3.9の実行例と同じように、FORMAT.EXE以下のファイルをコピーしてもよいでしょう。今回のものとまったく同じシステムディスクになります。

ここまでの実習では、MS-DOSのシステムディスクや、MS-DOSが組み込まれたビジネスソフトのシステムディスクなどをもとにして、それらをまるごとコピーしたディスクや、MS-DOSシステムだけをコピーしたMS-DOSシステムディスクを作成しました。

また、任意のファイルを選択してコピーし、必要なファイルだけを集めたディスクを作ることも行いました。つまり、これらの操作を組み合わせれば、どのような内容のバックアップコピーでも作ることができるようになったわけです(ただし、これまでに実習したことは、それぞれのコマンドの基本的な使い方であり、それぞれにはさらに多くの使い方があるが、それらは、これから徐々に実習していく)。

3.5　ビジネスソフトを利用している方へ

さて、本章の最後に、これまでのまとめの意味で、日常使用している各ビジネスソフトのシステムディスクを例にとって、これをもとにMS-DOSのシステムディスクを作ってみましょう。ここでは、MS-DOSが組み込まれたMultiplanの常用ディスクを例にしていますが、いずれの場合もFORMATコマンドのプログラムファイルFORMAT.EXE(または .COM)が含まれていることが前提です。図3.10に、リセットボタンによる起動から、システムディスクの完成までの一連の実行例を示します。作業内容は今までに実習したことばかりですので、疑問があれば、再度ページを戻して読み直してみてください。

Multiplanの常用ディスクをドライブA：にセットする。このディスクには、2章の図2.6に示した(1)(b)の場合の自動スタートが設定されている

(リセットボタンを押す)

```
NEC PC-9800 Series Personal Computer

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30
Copyright 1981,88 Microsoft Corp. / NEC Corporation

自由文変換システム　ＡＴＯＫ６ ver 1.2
Copyright 1986,87 (株)ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 3.30

A>ECHO OFF
現在の日付は 1989-07-19 (水) です.
日付を入力してください: ^C
```

……CTRL＋C を入力して、強制的にコマンドレベルに戻す

```
バッチ処理を中止しますか <Y/N> Y
```

……「Y」を入力して自動スタート機能を中止する

```
A>DIR ↵
```

……MultiplanのディスクからMS-DOSが起動し、コマンドレベルになった。このディスクの内容を確認するためにDIRコマンドを実行する

```
 ドライブ A: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは A:¥

COMMAND  COM     24931  88-07-13   0:00
FORMAT   EXE     97766  88-07-13   0:00
MP       COM     36210  87-07-12   3:10
MP       COD     71904  87-07-12   3:10
MP       SYS    132656  87-07-12   3:10
MP       DAT     29162  87-07-12   3:10
MP       HLP    102194  87-07-12   3:10
MP       PIF       369  87-07-12   3:10
MOUSE    SYS      4247  87-07-12   3:10
SETMP    INI        55  89-07-19  15:12
601_P    CPD      6460  87-07-12   3:10
601_DL   CPD     12112  87-07-12   3:10
601_DP   CPD     10294  87-07-12   3:10
601_L    CPD      6724  87-07-12   3:10
201H     CPD      7010  87-07-12   3:10
201H_D   CPD     13792  87-07-12   3:10
201H_DS  CPD     13698  87-07-12   3:10
201H_S   CPD      6932  87-07-12   3:10
ATOK6A   SYS     54271  87-06-15  12:00
ATOK6B   SYS     21372  87-06-15  12:00
ATOK     DIC    454144  87-06-15  12:00
CONFIG   SYS        78  89-07-19  15:12
AUTOEXEC BAT        27  89-07-19  15:12
MP       INI       258  89-07-19  17:24
         24 個のファイルがあります.
      34816 バイトが使用可能です.
```

（FORMAT EXE の行）……FORMATコマンドのプログラムファイル。のちほどこのファイルをCOPYコマンドで新しいディスクにコピーする

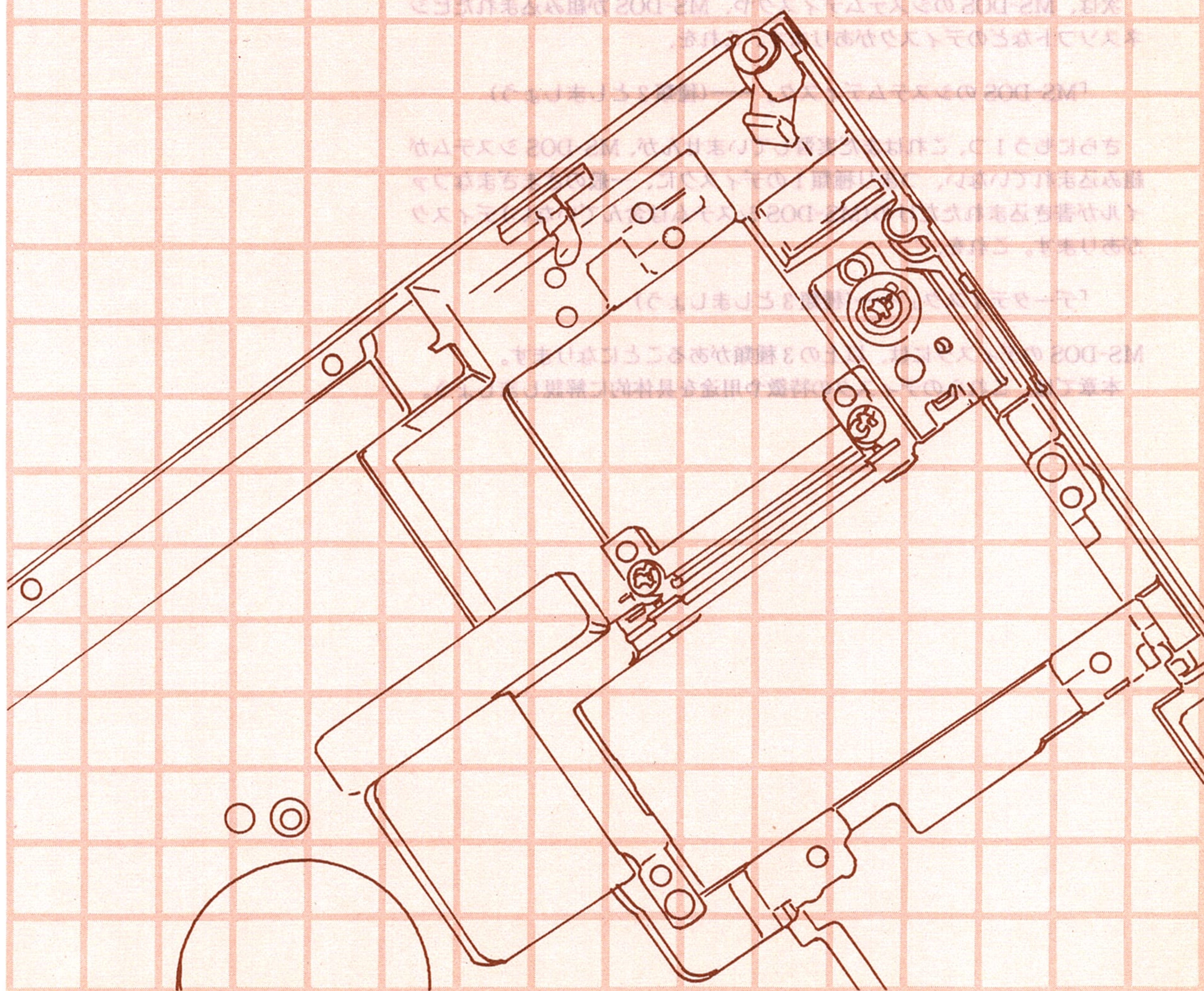
— 図3.10 — (次ページに続く)

```
A>FORMAT B:/S ⏎ ……………/Sスイッチを付けてFORMATコマンドを実行する
Format Version 4.10

新しいディスクをドライブ B: に挿入し
どれかキーを押してください x ……………実習用ディスクをドライブB:にセットし、適当なキーを入力する

ディスクのタイプは  1 : 640(KB)  2 : 1(MB)  = 2 ⏎ ……1Mタイプのディスクをフォーマット処理する

目的のディスクは 1MB FD です

フォーマット中です. . .   残り  152 トラックです
0  10  20  30  40  50  60  70  80  90  100

システムを転送しました

   1250304 バイト  全ディスク容量
    120832 バイト  システム領域
   1129472 バイト  使用可能ディスク容量

別のディスクをフォーマットしますか(Y/N) N ⏎ ……………FORMATコマンドを終了する
A>COPY A:FORMAT.EXE B: ⏎ ……………COPYコマンドで、FORMAT.EXEファイルをディスクB:側にコピーする
        1 個のファイルをコピーしました.

A>DIR B: ⏎ ……………新しく作成されたドライブB:上のシステムディスクの
                     内容をDIRコマンドで確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

COMMAND  COM    24931  88-07-13    0:00  完成したシステムディスクは、これだけのファ
FORMAT   EXE    97766  88-07-13    0:00  イルが格納されている(COMMAND.COM
                                         は、フォーマット処理時にコピーされている)
        2 個のファイルがあります.
  1031168 バイトが使用可能です.

A>
```

図 3.10　Multiplan の常用ディスクから MS-DOS システムディスクを作る一連の作業例

以上の作業で、MS-DOS のシステムディスクができました。コピー元にした Multiplan のシステムディスク(常用ディスク)には、普通、MS-DOS コマンドのプログラムファイルは、ほんの 2、3 のものしか含まれていないでしょう。しかし、本章で実習した程度のコマンドだけでも、十分に MS-DOS の基本を学ぶことができますので安心してください。

4章
システムディスクとデータディスク

4章 システムディスクとデータディスク

MS-DOSで使用するディスクは、中身に何が書き込まれているかによって、性格の異なるいくつかの種類に分けることができます。ディスクをコピーしたりファイルをコピーしたりする際には、それらの違いをよく理解した上で、用途に合ったディスクの使い方をしましょう。

ディスクは、その中身によって次のような種類に分けられます。

ディスクはまず最初に、「フォーマット処理」(初期化)と呼ばれるディスクの前処理を行わなければ使うことはできません。この処理を行った状態のディスクが、MS-DOSで使用する最も初期のディスクです。このディスクを、

「フォーマット処理されたままのディスク」——(種類1としましょう)

次は、MS-DOSのシステムディスクや、MS-DOSが組み込まれたビジネスソフトなどのディスクがあります。これを、

「MS-DOSのシステムディスク」——(種類2としましょう)

さらにもう1つ、これはまだ実習していませんが、MS-DOSシステムが組み込まれていない、つまり種類1のディスクに、一般のさまざまなファイルが書き込まれただけの(MS-DOSシステムは含んでいない)ディスクがあります。これを、

「データディスク」——(種類3としましょう)

MS-DOSのディスクには、以上の3種類があることになります。

本章では、これらのディスクの特徴や用途を具体的に解説しましょう。

4.1 フォーマット処理されたディスク〈種類 1〉

私たちは 3.1 節で、フロッピーディスクのフォーマット処理の実習を行いました。MS-DOS で使用するディスクの中で、最も初期状態にあるものが、このフォーマット処理をしたままの状態のディスクです(/S スイッチを付けずに行ったもの)。

どのような作業を行うにも、すべてこの処理をしたディスクが前提ですので、MS-DOS で使用するために購入したフロッピーディスクは、あらかじめ全部に、このフォーマット処理をしておくことをお勧めします(実際に使う直前になって、あわててフォーマット処理しなくてもよいように)。

さて、フォーマット処理されたディスクとは何なのか、次の模式図で示しましょう(**図 4.1**)。ディスクへの記録は、磁気円盤シートに同心円状に行われることは前にも述べましたが、ここではディスク上の記録領域を直方体で示しています。ディスクの最大記録容量が、この直方体の全体積と考えてください。具体的には、左端がトラックの外周部、右端が内周部に相当します。

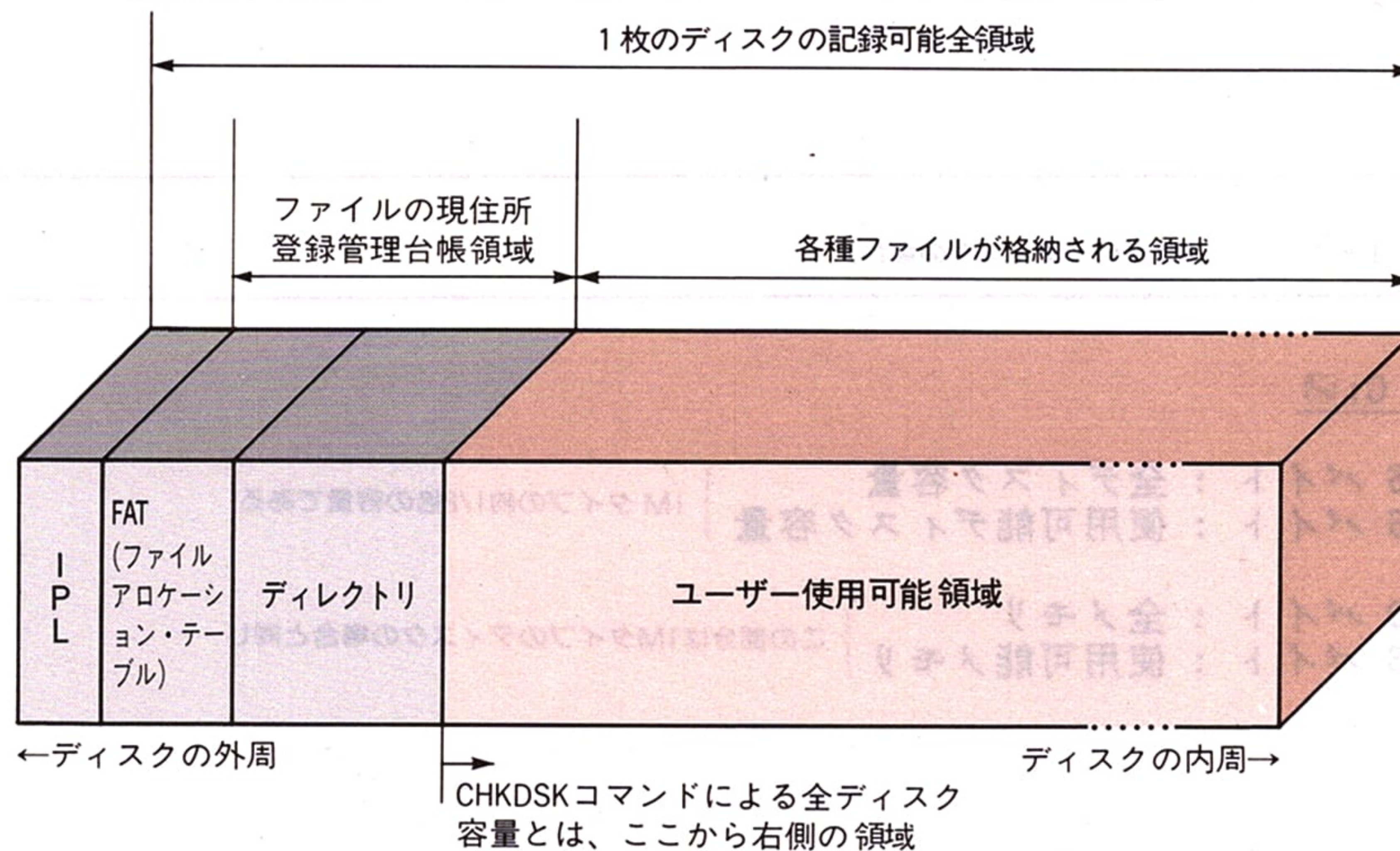



図 4.1　フォーマット処理されたディスクの内部を表した模式図

新品のディスクや、BASIC や CP／M など、ほかのシステムで使っていたディスクは、フォーマット処理を行うことによりディスク全面に MS-DOS 用のトラックが書き込まれ、MS-DOS が読み出し／書き込みできる形式のディスクとなります。そのとき同時に、図 4.1 に示されているような、リセットボタンや電源投入時に MS-DOS を起動させるための **IPL**(イニシャルプログラムローダ)と呼

ばれるプログラムや、**ファイルアロケーション・テーブル**(FAT：各ファイルの住所録)、**ディレクトリ**(各ファイルの登録管理台帳とでもいうべき領域)が作られます。MS-DOSは、ディスクに書き込まれるさまざまなファイルの管理情報を、この領域に格納し、それを参照したり更新したりして、それぞれのファイルの読み／書き／更新／削除などの管理を行っています。

この「種類1」のディスクは、フォーマット処理をしたままの状態であり、MS-DOSシステムやファイルなどは何も書き込まれていないため、MS-DOSで取り扱うディスクの中では最も大きな記憶容量があります。図4.1に示されている**ユーザー使用可能領域**には、MS-DOSシステム部を含め、いろいろな種類のファイルを自由に書き込むことができるわけです。

(5インチ2HDなどの1Mタイプのフロッピーディスクの場合)

```
A>CHKDSK B:⏎ ……ドライブB:にセットされているディスクに対して、CHKDSKコマンドを実行する

  1250304 バイト : 全ディスク容量          } 両者の値が一致している点に注目
  1250304 バイト : 使用可能ディスク容量    }

   655360 バイト : 全メモリ ……コンピュータのメインメモリが640Kバイトである。
                              (1K＝1024バイト。1024×640＝655360バイト)
   426624 バイト : 使用可能メモリ ……上のメモリの中でユーザーが使用可能な容量

A>
```

(5インチ2DDなどの640Kタイプのフロッピーディスクの場合)

```
A>CHKDSK C:⏎

   649216 バイト : 全ディスク容量          } 1Mタイプの約1/2倍の容量である
   649216 バイト : 使用可能ディスク容量    }

   655360 バイト : 全メモリ                } この部分は1Mタイプのディスクの場合と同じ
   426736 バイト : 使用可能メモリ          }

A>
```

図4.2　CHKDSKコマンドでディスクの記憶容量を調べる実行例

ディスクの容量は、FORMAT コマンドを実行したときにも表示されますが(図 3.1、3.2)、**図 4.2** の実行例に示すように「CHKDSK コマンド」を使って詳しく調べることができます。なお、CHKDSK コマンドを実行するには、ディスク上に、

CHKDSK.EXE(あるいは .COM)

という名前のプログラムファイルが存在していなければなりません。DIR コマンドで確認してください(CHKDSK コマンドの解説は、7.9 節で行う)。

この例では、フォーマット処理したままの、なにもファイルが書き込まれていないディスクをチェックしていますので、全ディスク容量(ファイルがなにも書き込まれていない空の状態)と、使用可能ディスク容量(今後使用できる空き領域)とが同じになっています。

次は、MS-DOS システムディスクの解説に移りましょう。

4.2　MS-DOS システムディスク〈種類 2〉

MS-DOS を起動できるディスクのことを **MS-DOS システムディスク**といい、その作成を 3.3 節で実習しました。

前項のフォーマット処理されたままのディスクは、MS-DOS で使用することはできますが、リセットボタンを押しても当然のことながら MS-DOS は起動しません。では、フォーマット処理されたままのディスクと、MS-DOS システムディスクとはどこが違うのでしょう。それは、3.3 節でも触れていますが、ディスク上に MS-DOS 自身のプログラムが、書き込まれているかいないかの違いです。

MS-DOS 自身は、3 つのプログラムファイルから構成されています(本体の 2 つのプログラムと COMMAND.COM との 3 つの部分に分れている)。それらを簡単にまとめたものを**表 4.1** に示しましょう。

	ファイル名	機能・実体	DIR コマンドによる表示
MS-DOS システム	IO.SYS	周辺装置の入／出力に関するプログラム	普通は表示されない
	MSDOS.SYS	MS-DOS の中心核	
	COMMAND.COM	入力コマンドの判別／実行に関するプログラム、および内蔵コマンドの全プログラム	表示される

表 4.1　MS-DOS システムを構成する 3 つのプログラム

フォーマット処理されたディスクに、表4.1に示した3つのファイルを正しくコピーすると、MS-DOSを起動できるMS-DOSシステムディスクができます。

では、MS-DOSシステムディスクの状態を、さきほどの図4.1と同じ模式図で示しましょう。たとえば、/Sスイッチ付きFORMATコマンドで作成したシステムディスク（前章の図3.5で実習している）などはこのような状態です（**図4.3**）。

1枚のディスクの記録可能全領域
MS-DOS起動用プログラム
ファイルの現住所登録管理台帳領域
MS-DOSシステムプログラム（計約120000バイト＊）
ユーザー使用可能領域
IPL
FAT（ファイルアロケーション・テーブル）
ディレクトリ
IO.SYS
MSDOS.SYS
COMMAND.COM
←ディスクの外周
ディスクの内周→
このCOMMAND.COMファイルは、ユーザー使用可能領域のどこにあってもよい。
＊この値は、MS-DOSバージョン3.3の例

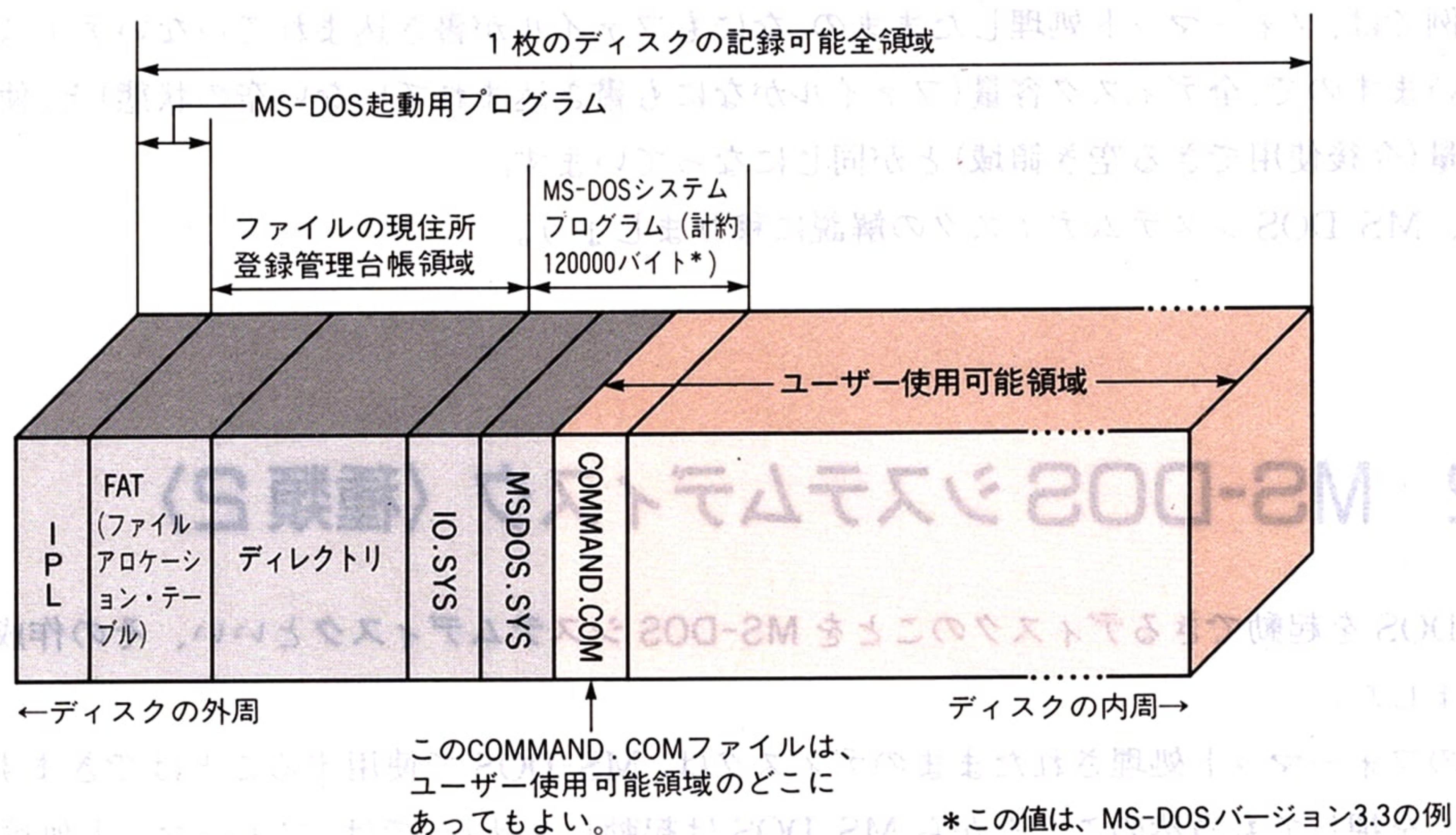



図4.3　MS-DOSシステムディスクの内部を表した模式図

これがMS-DOSシステムディスクの内部の状態です。図4.1のものと比較してください。図4.1のユーザー使用可能領域の先頭に、「IO.SYS」と「MSDOS.SYS」という2つのファイル（システムファイルと呼ばれる）が書き込まれています。そのため、ここでのユーザー使用可能領域はそれ以降となり、またその最初には「COMMAND.COM」が書き込まれています。

この「COMMAND.COM」ファイルは、ユーザー使用可能領域のどこに位置していてもよいのですが、FORMATコマンドの/Sスイッチで作成した場合は、ユーザー領域に書き込まれる最初のファイルとなりますので、その先頭に置かれることになります。

このディスクには、MS-DOSのシステムが書き込まれていますので、その分だけユーザーの使用可能領域が減少しています。そのことは、「FORMAT/S」コマンドの実行が終了したときの表示（図3.5参照）にも各容量が示されていますが、もう一度このディスクをCHKDSKコマンドで確認してみましょう（**図4.4**）。

5インチ2HDなど1Mタイプのフロッピーディスクの場合

```
A>CHKDSK B:⏎ ……………ドライブB:上のディスクに対してCHKDSKコマンドを実行する

  1250304 バイト : 全ディスク容量
    95232 バイト : 2 個のシステムファイル ……………IO.SYSとMSDOS.SYSの2つのファイル*
    25600 バイト : 1 個のユーザーファイル ……………COMMAND.COMのこと*
  1129472 バイト : 使用可能ディスク容量 ……………1250304−(95232+25600)=1129472*

   655360 バイト : 全メモリ
   426624 バイト : 使用可能メモリ

A>                                    *それぞれの容量は、MS-DOSバージョン3.3の例
```

図 4.4 ディスクの記憶容量を調べる CHKDSK コマンドの実行例

このように、フォーマット処理したままのディスクと比べて、3 つのファイルから成る MS-DOS システムが占める容量分(この例では 120832 バイト)だけ、私たちが使用できる領域が減少しています。しかしこのディスクは、MS-DOS を起動することができます。したがって、MS-DOS システムが組み込まれていない各種のアプリケーションソフトのディスクから、必要なファイルをこのディスク上にコピーすることにより、日常の作業に使う、MS-DOS システムを組み込んだ常用ディスクを作ることができるわけです。さらに、10 章で解説する「自動スタート」の手続きをすれば、ディスクをセットしてリセットボタンを押すだけで、自動的に目的のプログラムを起動させることもできます(1.1 節で示した Multiplan の例のように)。

■ 各メーカーから提供されている MS-DOS システムディスク

MS-DOS システムディスクと呼べる最小限のものは、まずリセットボタンで MS-DOS を起動でき、表 4.1 に示した 3 つのファイルが存在している状態のディスクです(ただし、DIR コマンドでディスク上のファイルを見たときは、「COMMAND.COM」だけが表示される)。

このディスクだけでも、かなりいろいろな MS-DOS コマンドが使えます。何ができて何ができないかは 8 章で解説しますが、これまでの知識でも、DIR、COPY などのコマンドは使え、FORMAT や SYS、DISKCOPY などのコマンドは使えないことがわかると思います。つまり、ディスク上にそのコマンドのプログラムファイルが必要なコマンドは使えないわけです。

さて、各メーカーが提供している MS-DOS システムディスクとはどういうものなのか、その内容を見ておきましょう。2.2 節でも行っていますが、ここでもう一度、DIR コマンドで全部のファイルを表示してみましょう(図 4.5a)。

（PC-9800シリーズ用MS-DOSシステムディスク（バージョン3.3）の3枚組のフロッピーディスクのうちの＃1のもの）

```
A>DIR/W ↵

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM  ADDDRV   EXE  ASSIGN   EXE  ATTRIB   EXE  APPEND   COM
BACKUP   EXE  CHKDSK   EXE  CUSTOM   EXE  DELDRV   EXE  DISKCOPY EXE
DUMP     EXE  EDLIN    EXE  FC       EXE  FORMAT   EXE  KEY      EXE
MENU     COM  MENUED   EXE  MSASSIGN COM  RENDIR   COM  REPLACE  EXE
RESTORE  EXE  SPEED    EXE  SWITCH   EXE  SYMDEB   EXE  SYS      EXE
TREE     EXE  USKCGM   EXE  XCOPY    EXE  EMSDRIVE SYS  GRAPH    LIB
GRAPH    SYS  MOUSE    SYS  NECAIK1  DRV  NECAIK2  DRV  PRINT    SYS
RAMDISK  SYS  RSDRV    SYS  BATKEY   COM  CHKFIL   EXE  CHKENV   COM
SETUP    EXE  SETUP2   EXE  AUTOEXEC BAT  HDCOPY   BAT  HDFORM   BAT
MPX      BAT  OFGX     BAT  CONFIG   SYS  MENU     MNU  SAMPLE   MNU
SETUP    DEF  MP       INI  OFG      INI  TELE     INI  MENU     1
MENU     2    README   DOC
         57 個のファイルがあります.
     26624 バイトが使用可能です.

A>
```

図4.5a　メーカーが提供しているMS-DOSシステムディスクの内容（3枚組のうちの1枚）

このディスクでもやはり「COMMAND.COM」は、最初に位置していますね。システムディスクには、DIRコマンドで表示されるファイルとして、この「COMMAND.COM」ファイルが必要であることは何度も述べています。では、COMMAND.COM以外のたくさんのファイルは何に使うのでしょう。これらのファイルの主なものを、用途別に分類整理してみましょう（図4.5b）。

（図4.5aで表示されているMS-DOSシステムディスクの主要ファイルの用途）

MS-DOSおよびMS-DOSシステム周辺に関する操作や設定、保守のためのプログラムやファイル

ファイル	用途	備考
COMMAND COM	入力コマンド判別実行処理および内蔵コマンドの各種プログラム部	
AUTOEXEC BAT	MS-DOS起動時に実行されるバッチファイル（10章参照）	
CONFIG SYS	MS-DOSのシステム構築用ファイル（APPENDIX参照）	
MOUSE SYS	マウス（入力装置）を動作させるためのプログラム	デバイスドライバと呼ばれるファイル（PC-9800シリーズ独自のもの）
PRINT SYS	プリンタを動作させるためのプログラム	〃
RSDRV SYS	RS-232Cインターフェイスを動作させるためのプログラム	〃
EMSDRIVE SYS	EMSを動作させるためのプログラム	〃
RAMDISK SYS	RAMディスクを動作させるためのプログラム	〃
SYS EXE	システムコピープログラム	
APPEND COM	データファイルの検索パスを指定するプログラム	
MSASSIGN COM	MS-DOS標準の周辺装置割り当てプログラム	
ADDDRV EXE	キャラクタ系デバイスドライバを組み込むプログラム	
DELDRV EXE	ADDDRVコマンドで組み込んだデバイスドライバを削除するプログラム	

— 図4.5b —（次ページに続く）

分類	ファイル名	拡張子	用途
NECが独自に用意した日本語入力システム(FEP)に関するプログラムやファイル	NECAIK1	DRV	NEC独自のFEPを動作させるためのプログラム
	NECAIK2	DRV	
	USKCGM	EXE	外字作成プログラム
ディスクやファイルの操作に関するプログラム	ATTRIB	EXE	ファイルの属性の設定や表示をするプログラム
	CHKDSK	EXE	ディスクのチェックおよびエラー修復プログラム
	DISKCOPY	EXE	ディスクをまるごとコピーするプログラム
	FORMAT	EXE	ディスクフォーマットプログラム
	FC	EXE	2つのファイルの内容比較プログラム
	TREE	EXE	指定したドライブのディレクトリとサブディレクトリ上のファイルなどを表示するプログラム
	XCOPY	EXE	サブディレクトリを含めたファイルとディレクトリをコピーするプログラム
	BACKUP	EXE	ハードディスクのバックアップコピーを行うためのプログラム
	RESTORE	EXE	BACKUPコマンドで作成されたバックアップファイルから、もとのファイルを復元するプログラム
ソフトウェア開発用のプログラム	EDLIN	EXE	エディタプログラム
	DUMP	EXE	ファイルダンププログラム
	SYMDEB	EXE	8086/8088用シンボリックデバッガ
NECが独自に用意したPC-9800シリーズのMS-DOS用のプログラムやファイル	ASSIGN	EXE	MS-DOSの論理装置に実際の周辺装置を割り当てるプログラム
	SPEED	EXE	RS-232Cインターフェイスの設定プログラム
	SWITCH	EXE	PC-9800のメモリスイッチの設定プログラム
	RENDIR	COM	ディレクトリ名を変更するプログラム
	KEY	EXE	ファンクションキーや特殊キーの設定プログラム
	README	DOC	TYPEコマンドで読む、お知らせファイル

図 4.5b　市販の MS-DOS システムディスクに含まれる主なファイルの用途別分類

このように分類すると、システムディスクに含まれる各種ファイルの用途が、おおよそ理解できると思います。

■ MS-DOS 組み込みの各種アプリケーションソフトのディスク

ここまで読み進んでこられたみなさんは、システムディスクの概念が徐々に理解されてきたと思います。MS-DOS が組み込まれたビジネスソフトなども、DIR コマンドでその内容を見れば、MS-DOS システムに関するファイルと、個々のソフトに関するファイルとを区別することができるでしょう。

ここで、MS-DOS が組み込まれたビジネスソフトの例として、Multiplan の常用ディスク(購入したときのオリジナルの状態ではなく、日常、仕事に使っているディスクの一例)に含まれるファイルを分類してみましょう(図 4.6)。

（PC-9800シリーズ用のMultiplanの常用ディスクのファイルを用途別に並べ換えたもの）

番号	ファイル名	拡張子	内容
①	COMMAND	COM	コマンドプロセッサ
	CONFIG	SYS	MS-DOSのシステム構築用ファイル（APPENDIX参照）
	AUTOEXEC	BAT	自動スタート用バッチファイル（10章参照）
	MOUSE	SYS	マウス（入力装置）を動作させるためのプログラム
②	FORMAT	EXE	フォーマットプログラム（MS-DOSコマンドのプログラムファイル）
③	ATOK6A	SYS	日本語入力システム（ATOK）関係のファイル
	ATOK6B	SYS	
	ATOK	DIC	
④	MP	COM	Multiplanのプログラム
	MP	COD	
	MP	SYS	
	MP	DAT	
	MP	HLP	
	MP	PIF	
	MP	INI	
	SETMP	INI	
	601_P	CPD	各機種に対応したプリンタ出力関係のファイル
	601_DL	CPD	
	601_DP	CPD	
	601_L	CPD	
	201H	CPD	
	201H_D	CPD	
	201H_DS	CPD	
	201H_S	CPD	

図 4.6　MS-DOS が組み込まれた Multiplan の常用ディスクに含まれるファイルの分類

この常用ディスクは、図 4.6 に示されているように、① MS-DOS 自身および MS-DOS のシステム関係のファイル、② MS-DOS コマンドのプログラムファイル、③日本語入力システム関係のファイル、それに④ Multiplan 関係のファイル、の 4 つのグループから構成されています。

この例からも、MS-DOS システムディスク上の各種のファイルの関係などが、ある程度理解できるのではないかと思います。

4.3 データディスク〈種類3〉

フォーマット処理したディスクにMS-DOSシステムをコピーせず、そのまま各種のファイルを書き込んだディスクは**データディスク**と呼ばれ、システムディスクとは次のような点で異なります。

- データディスクはMS-DOSを起動できない(当然だが)。
- データディスクに、あとからMS-DOSシステムをコピーすることはできない。
- ユーザーが使える使用可能容量は、システムディスクに比べ、MS-DOSシステムが占める容量分(MS-DOSバージョン3.3の場合、約120Kバイト)多い

では何か1つ、実際にデータディスクを作ってみましょう。市販のMS-DOSシステムディスクの中から、今までに登場したMS-DOSコマンドの4つのプログラムファイルを、フォーマット処理されたままの空のディスクにコピーします。その実行例を示しましょう(**図4.7**)。

```
A>COPY CHKDSK.EXE B:⏎
        1 個のファイルをコピーしました.

A>COPY DISKCOPY.EXE B:⏎
        1 個のファイルをコピーしました.

A>COPY FORMAT.EXE B:⏎
        1 個のファイルをコピーしました.

A>COPY SYS.EXE B:⏎
        1 個のファイルをコピーしました.

A>DIR B:⏎

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

CHKDSK   EXE    10384  88-07-13   0:00
DISKCOPY EXE    19604  88-07-13   0:00
FORMAT   EXE    97766  88-07-13   0:00
SYS      EXE    25480  88-07-13   0:00
        4 個のファイルがあります.
  1094656 バイトが使用可能です.

A>
```

COPYコマンドの実行。ドライブA:のそれぞれのファイルをドライブB:にコピーする

DIRコマンドを実行して、ドライブB:にコピーされたファイルを確認する

上のCOPYコマンドの実行によって、この4つのファイルがコピーされている

図4.7　空のディスクに4つのファイルをコピーする実行例

これで、ドライブB：にセットした空のディスクに、4つのファイルがコピーされました。このディスクにはMS-DOSシステムはコピーされていません。つまり「データディスク」であるわけです。例によって、このディスクの内部を模式図で示しましょう(図4.8)。

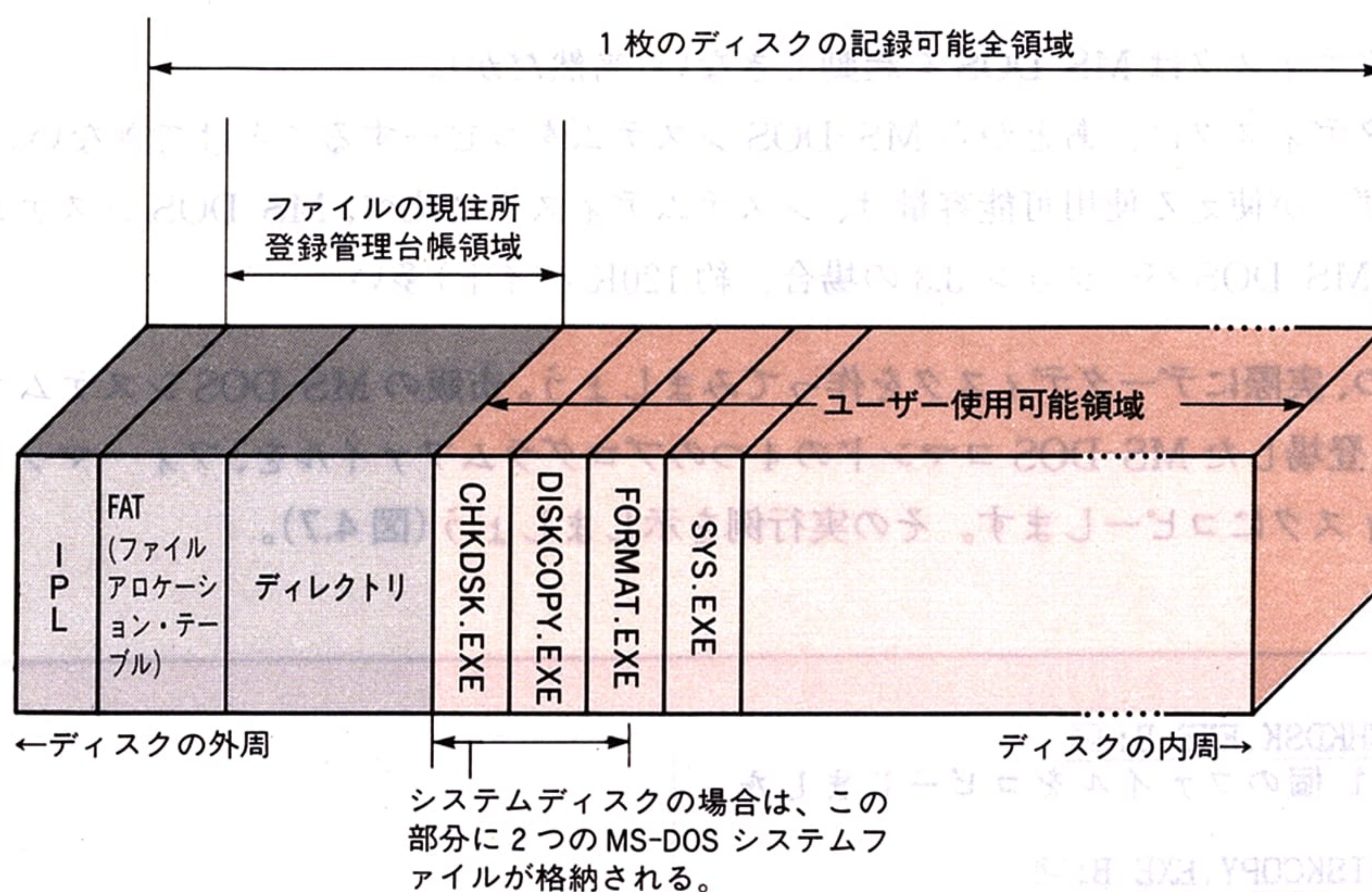



図4.8　4つのファイルがコピーされたデータディスクの内部を表した模式図

このディスクには、もしMS-DOSのシステムディスクであれば、「IO.SYS」「MSDOS.SYS」の2つのシステムファイルが書き込まれる位置に、「CHKDSK.EXE」以下のファイルが書き込まれています。ところがMS-DOSシステムディスクは、ユーザー使用可能領域の先頭に、その2つのシステムファイルを置かなければMS-DOSを起動することができません。――ここのところは重要です。つまりこのディスクは、MS-DOSシステムを書き込むべき領域を、ほかのファイルに占領されているわけです。もしこのディスクにMS-DOSシステムをコピーしようとすると、図4.9のようなエラーメッセージが表示され、コピーは拒否されます。どうしてもMS-DOSシステムをコピーしたいのであれば、すべてのファイルを削除するか、あるいはもう一度フォーマット処理をするしかありません。

```
A>SYS B:⏎ ……………SYSコマンドを実行して、ドライブA:のMS-DOSシステムを、ドライブB:のデータディスクに
                 コピーしようとする
ＳＹＳコマンド　Version 1.1
Copyright (C) 1988 NEC Corporation

受け側ディスクにシステム領域がありません　コピーは不可能であり、エラーメッセージを
                                        表示してコマンドレベルに戻る
A>
```

図 4.9　データディスクに MS-DOS システムはコピーできない

■ MS-DOS システムディスクに市販アプリケーションソフトを組み込む

最近の各種のアプリケーションソフトには、MS-DOS システムが含まれていないものが多くなりました(両者の提供ルートを分離し、責任を明確にする方針から)。それらのソフトは、MS-DOS システムがコピーされていない、つまりデータディスクの形式で提供されています。

したがって、それらのアプリケーションソフトを実行するには、MS-DOS システムディスクが必要になりますが、日常の仕事に使うには、MS-DOS システムをコピーした別のディスクに、使用するアプリケーションソフトの中から必要なファイルをコピーしておくと便利です。つまり、MS-DOS 組み込みの各種アプリケーションソフトの常用ディスクを作るわけです。さらに必要であれば、目的のプログラムを、リセットボタンや電源 ON による自動スタートで起動させることも可能です。

では、MS-DOS システムディスクに市販アプリケーションソフトを組み込む実例を示しましょう。アプリケーションソフトとして、エディタプログラム「Mifes-98」を例に、その中の必要なファイルをシステムディスク上にコピーし、MS-DOS 組み込みの「Mifes 常用ディスク」を作ってみます(**図 4.10a、図 4.10b**)(これをさらに自動スタートにする実習は 10.2 節で行う)。

```
ドライブA:にMS-DOSの常用システムディスク、B:に実習用ディスクをセットして実行する
A>FORMAT B:/S⏎ …………/SスイッチをFORMATコマンドを実行する。ドライブB:のディスクがフォーマット
Format Version 4.10        処理され、システムがコピーされる (3.3節で実習済)

新しいディスクをドライブ B: に挿入し
どれかキーを押してください x

ディスクのタイプは　1 : 640(KB)　2 : 1(MB)　=　2⏎

目的のディスクは 1MB FD です
　　　　　　　⌇
```

— 図 4.10a —（次ページに続く）

```
システムを転送しました

   1250304 バイト  全ディスク容量
    120832 バイト  システム領域
   1129472 バイト  使用可能ディスク容量

別のディスクをフォーマットしますか(Y/N) N↵ ……FORMATコマンドを終了する。
A>COPY ATOK6A.SYS B:↵                        ドライブB:のディスクがフォーマット処理され、
         1 個のファイルをコピーしました.        さらにMS-DOSシステムがコピーされた

A>COPY ATOK6B.SYS B:↵
         1 個のファイルをコピーしました.  ……日本語入力システムの各種のファイルをコピーする

A>COPY ATOK.DIC B:↵
         1 個のファイルをコピーしました.
A>COPY MIFES.EXE B:↵
         1 個のファイルをコピーしました.

A>COPY MIFES.HLP B:↵
         1 個のファイルをコピーしました.  ……ドライブA:のディスクをMifes-98のディスクと
                                            入れ換えてから、必要なファイルをドライブB:上
A>COPY AUTOEXEC.BAT B:↵                     にコピーする
         1 個のファイルをコピーしました.

A>COPY CONFIG.SYS B:↵
         1 個のファイルをコピーしました.

A>DIR B:↵ ……コピー先のディスクファイルを確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

COMMAND  COM     24931  88-07-13   0:00 ……FORMAT/Sコマンドによってコピーされたもの
ATOK6A   SYS     54271  87-06-15  12:00
ATOK6B   SYS     21372  87-06-15  12:00
ATOK     DIC    454144  89-07-30  16:50
MIFES    EXE     98358  88-11-02   4:11 ……COPYコマンドによってコピーされたもの
MIFES    HLP     42408  88-11-02   4:11
AUTOEXEC BAT       116  88-09-03   4:11
CONFIG   SYS        52  88-09-03   4:11
         8 個のファイルがあります.
   454656 バイトが使用可能です.

A> ……以上でMS-DOSシステムが組み込まれたMifes-98のシステムディスクが完成した
```

図 4.10a　MS-DOS が含まれていないアプリケーションソフトを、MS-DOS システムディスクに組み込む実行例

できたディスクをドライブA：にセットして、Mifes-98を立ち上げてみる

リセットボタンを押す

```
NEC PC-9800 Series Personal Computer

ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30
Copyright 1981,88 Microsoft Corp. / NEC Corporation

自由文変換システム　ＡＴＯＫ６ ver 1.2
Copyright 1986,87 (株)ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 3.30

A>MIFES
```

MS-DOSが起動した

日本語入力システムが組み込まれた

Mifesを立ち上げるコマンド。このディスクでは自動スタートバッチファイル（AUTOEXEC.BAT）によって自動的に入力される

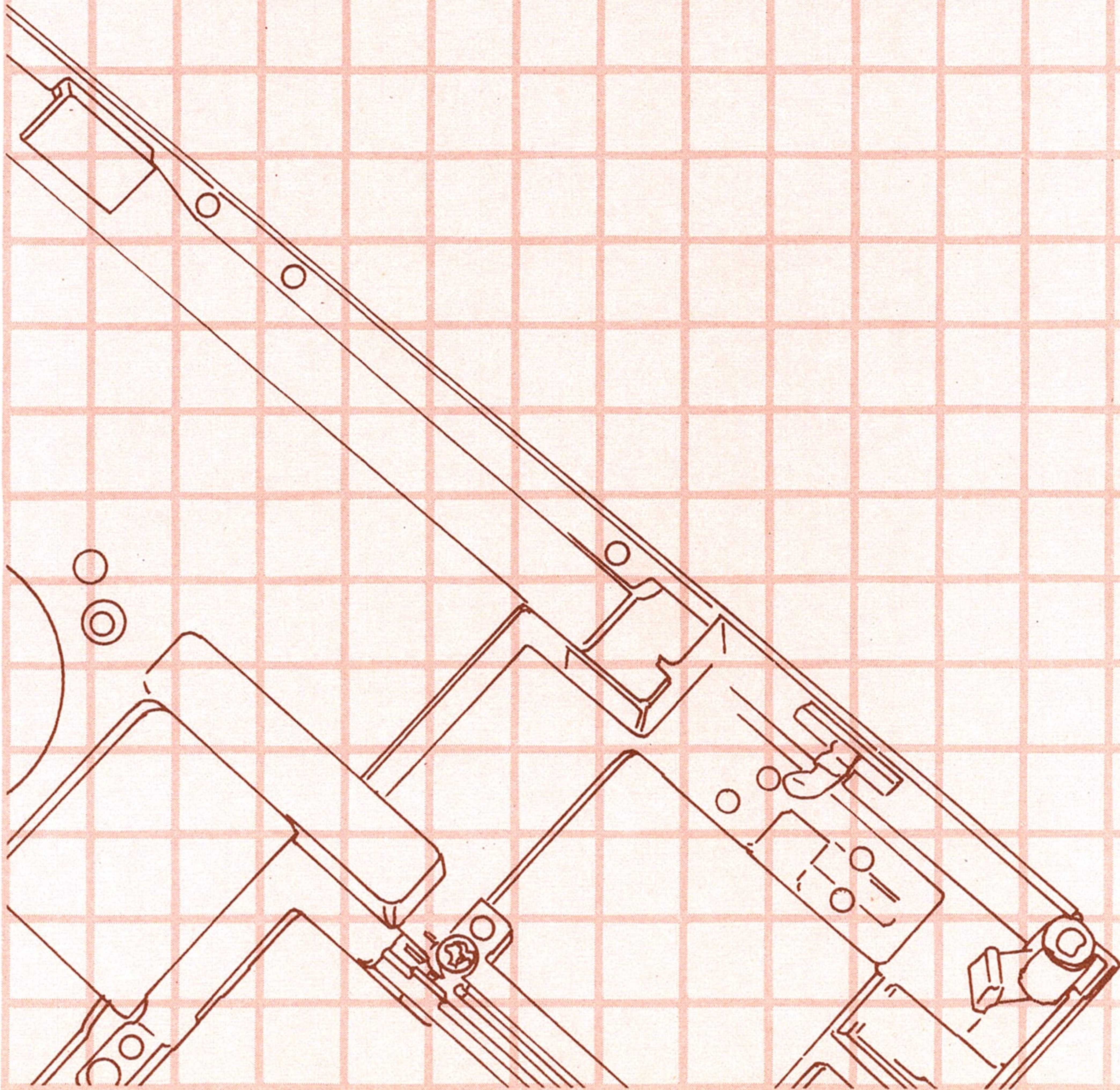
```
編集したいファイル名を入力して下さい [テキスト番号: 0]

カレントディレクトリー＝A:¥
  PATH =  A:¥*.*
A4.MIL         ADDDEC.MIL     ADDHEX.MIL     APPEND.MIL
AUTOEXEC.BAT   CALC.MIL       CHILDWIN.MIL   COMMAND.COM
COMPILE.MIL    CONFIG.SYS     CONTWID.MIL    DELNUM.MIL
FILTER.MIL     FINDWID.MIL    INITC.MIL      ISEARCH.MIL
KEISEN.MIL     LAYOUT.MIL     MACRO.SRC      MAKELOGO.MIL
MASM.MIL       MI.BAT         MIBACKUP.BAT   MIDEL.EXE
MIFES.EXE      MIFES.HLP      MIFES.LIB      MIFIND.EXE
MIFORM.EXE     MIFORM.MAC     MIKEY.DEF      MIKEY.EXE
```

Mifes-98が立ち上がり、編集するファイル名の入力を要求している

```
*.*CR  *.cCR  *.asm  A:CR  B:CR    *.*  *.c  *.asm  C:CR  D:CR
```

図 4.10b　作成した MS-DOS 組み込みのアプリケーションソフトのディスクを起動し、そのソフトを実行する

さて、MS-DOS上で使用するディスクの、ソフトウェア的な形態の違いについてひととおり解説してきましたが、私たちが日常使っている各種のアプリケーションソフトの常用ディスクのほとんどは、MS-DOSシステムを含んだ「種類2」の形態でしょう。また、ワープロの文書ファイルや、表計算ソフト、データベースソフトなどのデータファイルを保存するためのディスクは、MS-DOSを含まない「種類3」の形態でしょう。本章の知識は、すべての作業に関係する大切な基礎になりますので、十分理解しておいてください。

5章
コントロールキーによる便利な機能

私たちがMS-DOS上でいろいろな仕事を行う場合、何をするにもまず知っておかなければならない、コントロールキーによる便利な機能があります。それらの機能は、日常よく利用するものであり、どれもコントロール・キャラクタ(後述)の1文字をキー入力するだけの非常に簡単なものですので、誰にでもすぐに役立てることができます。

とくにコマンドの実行をキャンセルしたり、画面の表示(スクロール)を一時ストップする機能などは、しばしば使うことになるでしょう。

「コントロール・キャラクタ」というのは、普通、キーボードの左端にあるコントロールキー(CTRLキー)を押しながら、ほかのキーを押す場合に入力される「文字」(キャラクタ)のことをいい、たとえばCTRL＋Sと記されたものは、CTRLキーを押しながらSキーを入力することを意味します。これを「コントロール・エス」と読んでください。

5.1 コントロール・キャラクタ、5つの主要な機能

私たちが、日常の作業を行う場合、とりあえず、次の5つのコントロール・キャラクタによる機能を知っておけばよいでしょう(**表5.1**)。

	コントロール・キャラクタによる機能	入力するコントロール・キャラクタ
1	入力した文字を1文字ずつ取り消す	CTRL + H あるいは BS、DEL キー
2	入力した文字列を1行分取り消す	CTRL + X あるいは CTRL + U (ただし CTRL + U は MS-DOS のバージョン 3.*x* 以上で動作する)
3	ディスプレイの表示、スクロールを一時ストップする 再度の入力または他のキー入力により表示を再開する	CTRL + S
4	ディスプレイに表示される文字を、同時にプリンタにも出力させる 再入力によりプリンタ出力が解除される	CTRL + P
5	プログラム(コマンド)の実行を中止し、MS-DOSのコマンドレベルに戻る	CTRL + C

表5.1　最もよく利用されるコントロール・キャラクタによる5つの機能

表5.1に示したコントロール・キャラクタによる各機能は、純粋なMS-DOS上では有効ですが、いかなる場合にも有効かというと、そうではありません。MS-DOS上で実行される各種のアプリケーションソフト(たとえばワープロや表計算ソフトなど)を立ち上げ、それぞれの世界へはいってしまってからは、有効の場合も無効の場合もあります。これは、個々のソフトウェアの作り方次第です。たとえば、ワープロソフトの「新松」における CTRL + P は、画面に表示される文字を同時にプリンタに出力する機能ではなく、削除した文字を復活させる機能に変えられているという具合です。

それでは、表5.1に示した各項目を、順に実習解説していきましょう。

5.2 ミスタイプ時の修正

ミスタイプの修正については2.3節でも解説しましたので、改めて取り上げる必要もないくらいですが、「コントロール・キャラクタによる機能」という面から、もう一度見てみましょう。

非常に簡単なことなので、実習は行いませんが、今回はミスタイプをした際の1文字削除に、BSキーや、DELキーを使わず、CTRL＋Hをキー入力してみてください。BSキーや、DELキーを使った場合と、まったく同じ結果が得られます(ブラインドタッチでキー入力する人に効果的)。

MS-DOSにとって、ミスタイプ時の1文字削除は、本来CTRL＋Hなのですが、頻繁に使われるキーであるため、専用特殊キーのBSやDELなどを、ソフトウェア的にCTRL＋Hとみなすように、各社のMS-DOSが作られているのです。

1行削除のCTRL＋XやCTRL＋Uの使い方も同じように簡単です。再度の実習は行いませんが、2.3節の図2.13を参考にしてください。

5.3 プリンタへの出力

MS-DOSでは、画面に表示されるいろいろな文字をプリンタに出力させることが簡単にできます。具体的には、CTRL＋Pのキー入力を境に、画面に表示される文字はすべて、同時にプリンタにも送られて印字されます。つまり、画面上に表示されるものが、そのまま同時にプリンタに打ち出されるわけです。ただし、その対象は「文字」に限られ、グラフィックス関係の図形などは含まれません。**図5.1**にその実行例を示しましょう。(リダイレクトの機能を使ってプリンタに出力することも可能であるが、それについては12章で解説する)

このように画面に表示されるものは、そのままプリンタに出力させることができます。たとえば、DIRコマンドやCHKDSKコマンドなど、実行前にCTRL＋Pを入力しておけば、画面の表示と同じ記録を紙の上に残すことができるわけです。また、7.4節で実習するTYPEコマンド(文字ファイルの内容を読み出して画面に表示するコマンド)なども、CTRL＋PをONにして実行すると、**文字ファイル**(文字データのみでできているファイル。**アスキーファイル**とか**テキストファイル**などと呼ばれる)の内容を簡単に印字することができます。

ディスプレイ上の表示

CTRL＋P 入力（プリンタ出力ON）

```
A>DIR/W ⏎
```
……ワイド表示の /Wスイッチを付けてDIRコマンドを実行する。ドライブA：には常用システムディスクがセットされている

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    CONFIG   SYS    ATOK6A   SYS    ATOK6B   SYS    ATOK     DIC
PRINT    SYS    RSDRV    SYS    MOUSE    SYS    FORMAT   EXE    DISKCOPY EXE
CHKDSK   EXE    SYS      EXE    FC       EXE    FIND     EXE    MORE     COM
SORT     EXE    TREE     EXE    ATTRIB   EXE    EDLIN    EXE    README   DOC
AUTOEXEC BAT
       21 個のファイルがあります.
   357376 バイトが使用可能です.
```

CTRL＋P 入力（プリンタOFF）

```
A>CHKDSK ⏎
```
……ドライブA：(システムディスクがセットされている)に対してCHKDSKコマンドを実行する

```
  1250304 バイト : 全ディスク容量
    95232 バイト : 2 個のシステムファイル
   797696 バイト : 21 個のユーザーファイル
   357376 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   454096 バイト : 使用可能メモリ

A>
```

そのときのプリンタの印字出力

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    CONFIG   SYS    ATOK6A   SYS    ATOK6B   SYS    ATOK     DIC
PRINT    SYS    RSDRV    SYS    MOUSE    SYS    FORMAT   EXE    DISKCOPY EXE
CHKDSK   EXE    SYS      EXE    FC       EXE    FIND     EXE    MORE     COM
SORT     EXE    TREE     EXE    ATTRIB   EXE    EDLIN    EXE    README   DOC
AUTOEXEC BAT
       21 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```
このように印字された。どの時点で CTRL＋P をONし、どの時点でOFFしたかに注目

図 5.1 CTRL ＋ P により、コマンドが実行される際の画面上の表示を同時にプリンタへ出力する

いったんONにされたこの機能は、再度 CTRL ＋ P を入力すれば、解除することができます(CTRL ＋ P を入力するたびに機能がON-OFF-ON-OFFされる)。その実行例をもう一例、図5.2に示しましょう。

ディスプレイ上の表示

CTRL＋P入力(出力ON)

```
A>↵
A>TYPE README.DOC↵

1.　　バックアップ作成時の注意


　ＰＳ９８－０１３(０１４)－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタ
フォーマット(７２０ＫＢ)でフォーマットされています。
ＰＳ９８－０１３(０１４)－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
　　　　　　　　　　　時ヶ／９スイッチを指定してください。
　なお、「スキップセクタ」が表示さ
に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけ
ます。

A>
```

↓

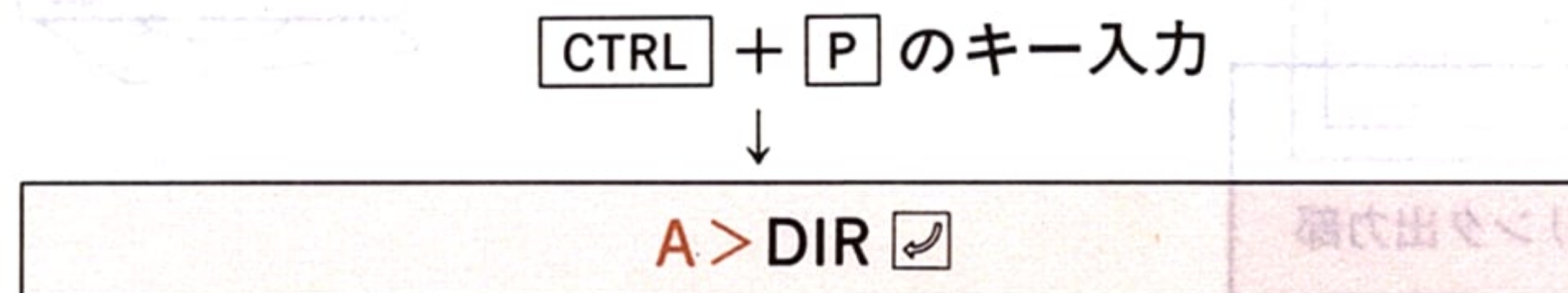
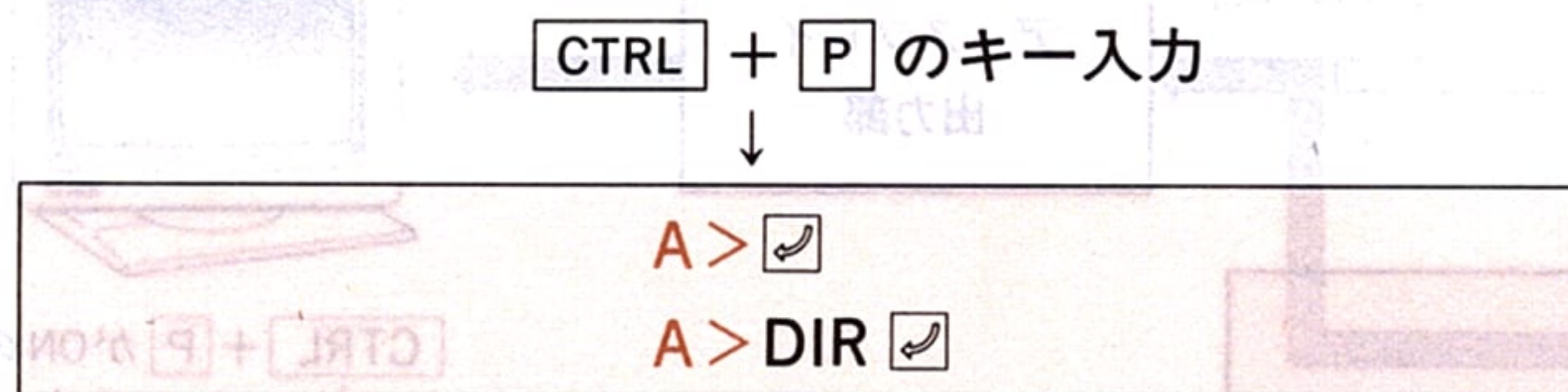
そのときのプリンタの印字出力

```
A>TYPE README.DOC
```
……今回はこのコマンドラインも印字されている

```
1.　　バックアップ作成時の注意


　ＰＳ９８－０１３(０１４)－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタ
フォーマット(７２０ＫＢ)でフォーマットされています。
ＰＳ９８－０１３(０１４)－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
　　　　　　　　　　　　　　スイッチを指定してください。
値がされていることを表しますので、
ご使用いただけます。

　なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品
に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけ
ます。

A>
```

図5.2　CTRL ＋ P の機能を使い、TYPEコマンドでテキストファイルをプリントアウトする

CTRL ＋ P の機能を使ってプリンタに印字させる場合には、どの時点で CTRL ＋ P をキー入力する(ON にする)かに注意してください。CTRL ＋ P がキー入力されたあとに画面に表示される文字がプリンタに送られるということを、よく考える必要があります。CTRL ＋ P のキー入力の前にすでに表示されてしまっている文字は、印字されません。したがって、

CTRL ＋ P のキー入力
↓
A＞DIR ↵

とした場合は、プロンプトの「A＞」の部分は、印字されません。もし、この部分も印字したいのであれば、

CTRL ＋ P のキー入力
↓
A＞↵
A＞DIR ↵

とすればよいのです(図 5.1、図 5.2 参照)。

もう 1 つ注意することは、多くのプリンタが印字を開始する条件は、復帰・改行コードを受け取るか、または、1 行の文字数が、規定数をオーバーしたときであることです(ページプリンタやレーザープリンタの場合は、改ページコードあるいは 1 ページの文字数がオーバーしたとき)。したがって、その条件を満たさない場合は、印字するデータをプリンタが受け取っても、印字は行われません。

次に、CTRL ＋ P のキー入力により、画面に表示されるデータが、同時にプリンタに送られる仕組みを図で示しておきましょう(**図 5.3**)。

注意：PC-9800 シリーズの MS-DOS バージョン 3.*x* では、プリンタを動作させる部分や、RS-232C インターフェイスを動作させる部分が MS-DOS から独立し、デバイスドライバという形態の別のプログラムになりました(バージョン 2.*x* のものは、「IO.SYS」の中に組み込まれている)。したがって、CONFIG.SYS ファイルの中で、プリンタのデバイスドライバを組み込む指示をしておかなければ、CTRL ＋ P は機能しませんので注意してください。具体的には、そのデバイスドライバのプログラムファイル「PRINT.SYS」をシステムディスク上に置き、

DEVICE＝PRINT.SYS

の 1 行を CONFIG.SYS ファイルに記述しておく必要があります。(CONFIG.SYS ファイルについては、APPENDIX 参照)。

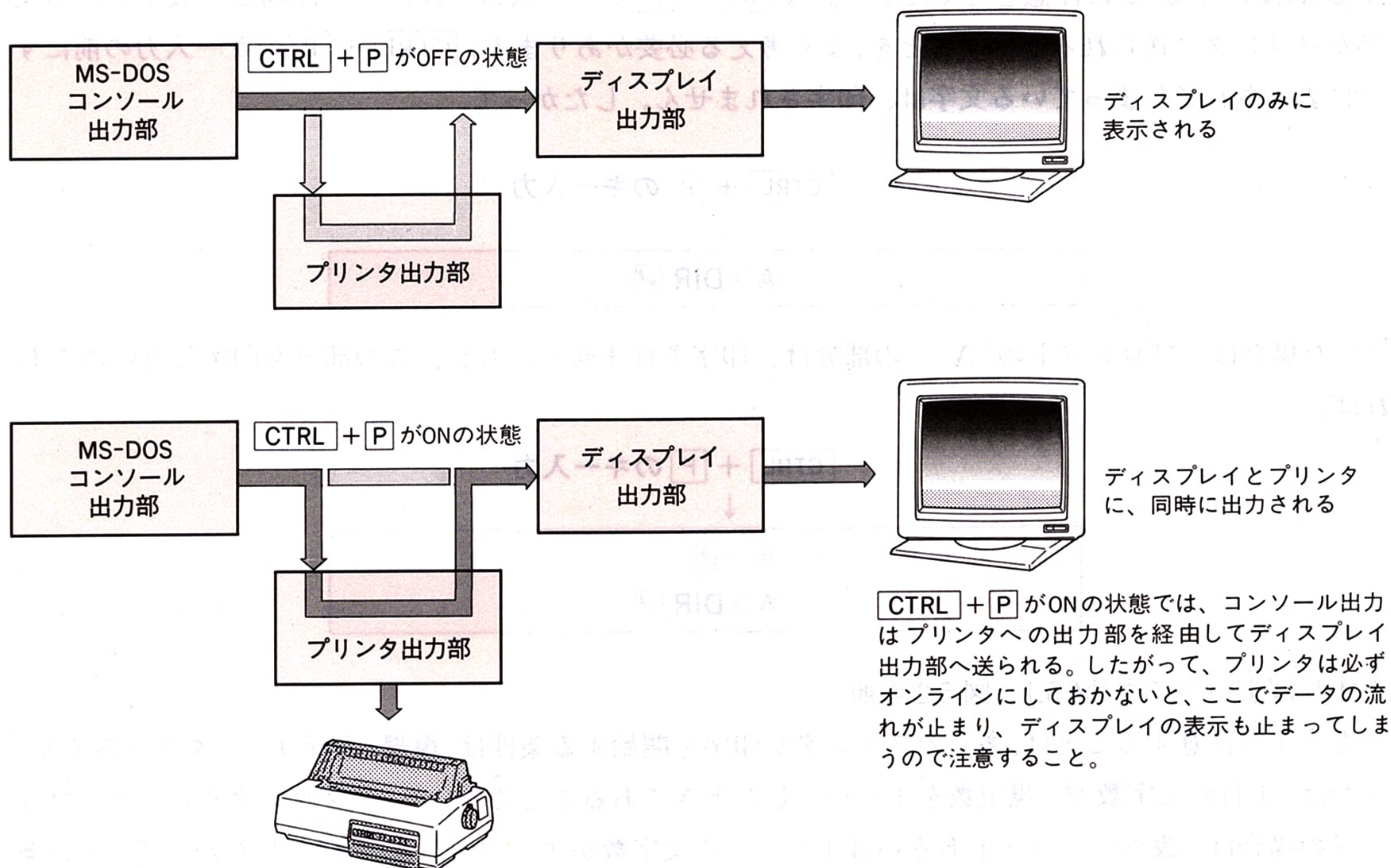



図 5.3　CTRL ＋ P による画面表示の印字出力 ON／OFF の仕組み

5.4 画面出力のポーズ

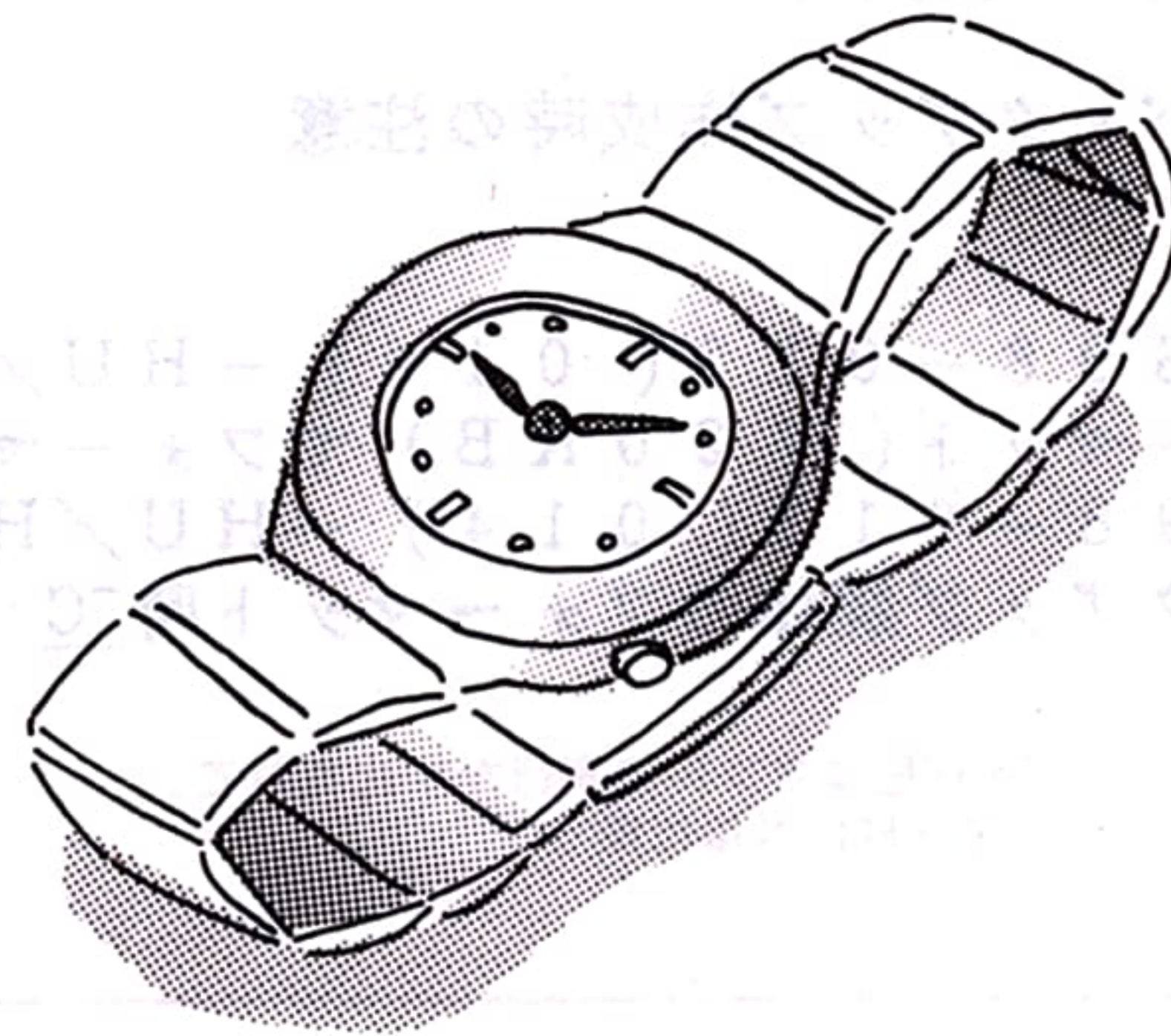
DIR コマンドでディスク上のファイル情報を見たり、TYPE コマンドで文書ファイルを読むような場合、表示データが1画面に収まらない場合は、表示がどんどんスクロールして(流れて)いきます。そのような場合、表示を一時ストップしなければ読むことはできません。そのために必要な、表示の一時ストップ機能が CTRL ＋ S です。CTRL ＋ S をキー入力した時点で、画面の表示がポーズ状態(一時ストップ状態)となります(図 5.4)。

このポーズ状態は、再度の CTRL ＋ S の入力か、または、ほかのキーの入力で解除されます。使い勝手としては、ほかのキーは使わずに、CTRL キーを押し続けながら、S キーだけで ON → OFF → ON → OFF するとよいでしょう。

```
A>TYPE README.DOC

 1.    バックアップ作成時の注意

     ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタ
   フォーマット（７２０ＫＢ）でフォーマットされています。
   ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
   バックアップ媒体のフォーマット時
```

この時点で表示がストップされている。再度のCTRL+Sの入力、またはその他のキーの入力でこのあとの表示が再開される

CTRL+Sの入力

図5.4　CTRL＋Sによる画面表示の一時ストップ

5.5　プログラム(コマンド)実行の中止

MS-DOS上でいろいろな仕事を行っていると、現在実行中の仕事を途中で中止したい場合がしばしばあります。そのようなとき、CTRL＋Cをキー入力することにより、現在実行中のプログラムを強制的に中止して、MS-DOSのコマンドレベル(A>、B>、などのプロンプトが表示される状態)に戻すことができます。ただし、いかなる場合にも、このCTRL＋Cの機能が有効であるわけではなく、それは実行中のプログラムが、CTRL＋Cによる機能を受け付ける状態にあるか、また、受け付けるように作られているかどうかによります。

この機能によるプログラム実行の「中止」は、一時お休みして、あとで再開するということではなく、「その場で打ち切り」(「異常終了」といってもよい)を意味します。途中で中止した場合、それまで行った仕事が無駄になるか(処理したデータなどが失われるか)、あるいはそこまでは救われるか(ちゃんとファイルに保存されているか)は、実行していたプログラムの作り方によりますので注意が必要です。

さて、CTRL＋Cの実習ですが、実はこの機能については、すでに2.2節と3.1節でも実習しています。再度ページを戻して、図2.8および図3.1、図3.3を参照してみてください。ここでは、もう1つの例として、さきの5.3節、5.4節で実習したTYPEコマンドをもう一度実行し、その実行途中でCTRL＋Cをキー入力して、その機能を確認してみましょう（CTRL＋PはONでもOFFでもかまわない）。TYPEコマンドにより文字ファイルが次々と表示されていく途中で、CTRL＋Cをキー入力すると、TYPEコマンドの実行が中止され、瞬時にコマンドレベルに戻ります。また、CTRL＋Sにより表示をポーズ状態にしておいたまま、CTRL＋Cをキー入力しても、結果は同じです。この実行例を図5.5に示します。

```
A>TYPE README.DOC

 1.    バックアップ作成時の注意


    ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタ
  フォーマット（７２０ＫＢ）でフォーマットされています。
  ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
  バックアップ媒体のフォーマット時^C

A>
```

^C：図5.4の実行例のCTRL＋Sにより表示がストップされている状態で、CTRL＋Cを入力した。表示が行われている途中で、CTRL＋Cを入力しても同様にコマンドの実行は中止される

A>：TYPEコマンドの実行が中止されて、コマンドレベルに戻った

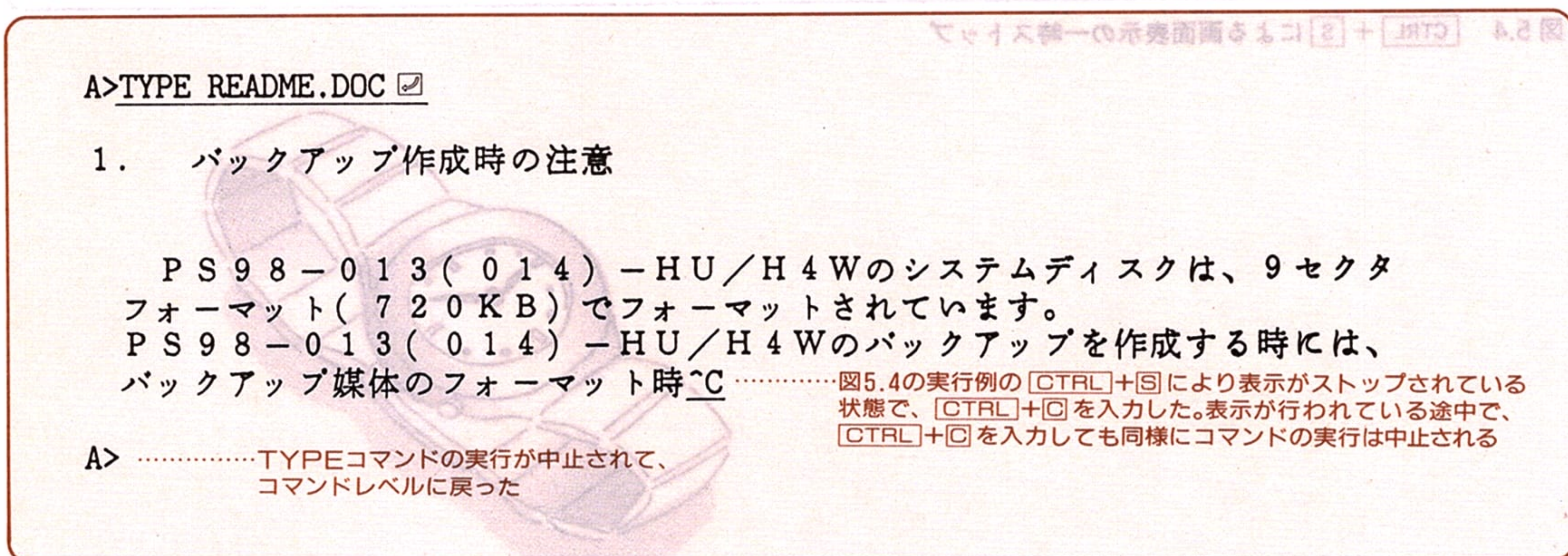
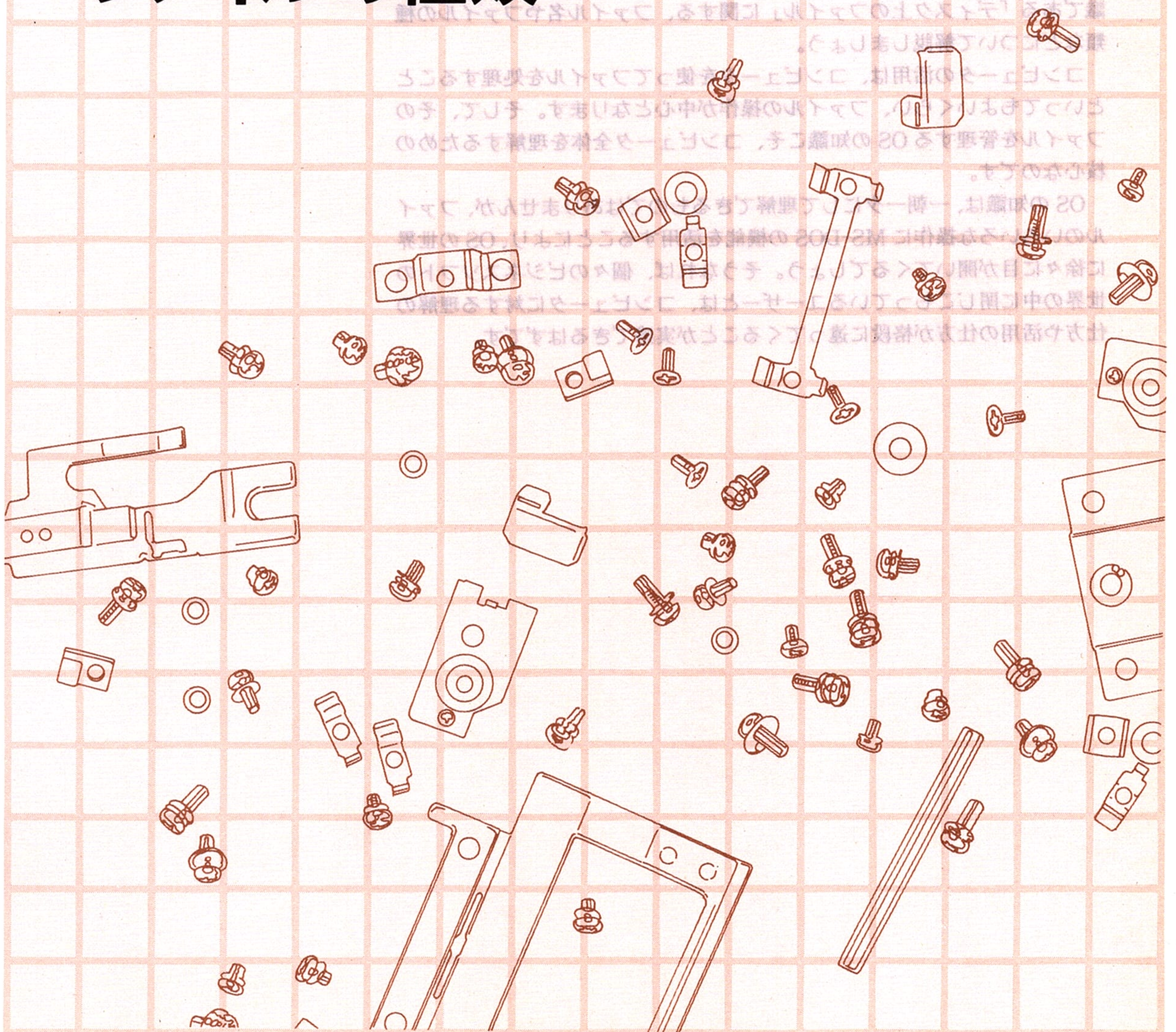



図5.5　CTRL＋CによりTYPEコマンドの実行を中止する

さて、MS-DOSの扱いに少し慣れたところで、次章では、実際の利用面で私たちに最も関係のある重要な知識、「ファイル」について解説しましょう。なお、MS-DOSの各種のコマンドの、いろいろな使い方については、そのあとの7章で実習解説します。

6章
ファイル／ファイル名／ファイルの種類

私たちが、MS-DOS上で行ういろいろな仕事は、ほとんどが「ファイル」をもとに行われています。マシン語のプログラムも、各種のデータや文書も、すべてはファイルという形式で取り扱われ処理されます。さらに言えば、MS-DOSでは、キーボードやディスプレイやプリンタやRS-232Cインターフェイスなどの周辺装置までも、「ファイル」という概念で取り扱われるのです。

ファイルとは何か——そのファイルの概念を理解することは、OSの概念を理解することにも等しく、広い基礎知識が必要とされます。したがって本章では、広義の「ファイル」についてではなく、とりあえず必要な知識である「ディスク上のファイル」に関する、ファイル名やファイルの種類などについて解説しましょう。

コンピュータの活用は、コンピュータを使ってファイルを処理することといってもよいくらい、ファイルの操作が中心となります。そして、そのファイルを管理するOSの知識こそ、コンピュータ全体を理解するための核心なのです。

OSの知識は、一朝一夕にして理解できるものではありませんが、ファイルのいろいろな操作にMS-DOSの機能を活用することにより、OSの世界に徐々に目が開いてくるでしょう。そうなれば、個々のビジネスソフトの世界の中に閉じこもっているユーザーとは、コンピュータに対する理解の仕方や活用の仕方が格段に違ってくることが実感できるはずです。

6.1 ファイル

本章の冒頭でも述べたように、OSの世界では**ファイル**という言葉の意味する範囲は広く、これを理解することは、OSの概念を理解することにも等しい知識が必要です。また、MS-DOSでは、キーボードやディスプレイなどの装置も「ファイル」と考えることができます(9章で解説)。つまり「ファイル」とは、ディスク上に記録されるデータの集まりというような物理的な意味だけでなく、コンピュータが管理するデータの集まりという概念で捉えなければなりません。

そのような広義のファイルの概念については、本書の全体を通して理解していくとして、ここではとりあえず、ディスク上のファイルについて解説しましょう。その一例として、パーソナル・コンピュータ上のワープロで、文書を作成する場合について考えてみます。

私たちがワープロの作業を行う際の手順は次のようなものでしょう。

①ワープロソフトの常用ディスク(MS-DOSが組み込まれている)をセットしてリセットボタンを押すと、まずディスク上の「MS-DOSのシステムファイル」がコンピュータ本体のメモリに読み込まれ、MS-DOSが起動する。同時に「CONFIG.SYSファイル」の内容が読み出され、その中で指示されている日本語入力システム*の「デバイスドライバ」(プログラムファイルの一種)がメモリに読み込まれる

②次に、ワープロソフトを自動的に立ち上げるための「自動スタートバッチファイル」(AUTOEXEC.BATファイルのこと。10.2節で解説)の内容が読み出される

③次に、そのバッチファイルの内容に従って、「ワープロソフト本体のプログラムファイル」がメモリに読み込まれ、ワープロが立ち上がる

④文書の作成中は、目的の漢字や熟語などを捜し出すために「辞書ファイル」が盛んに検索される

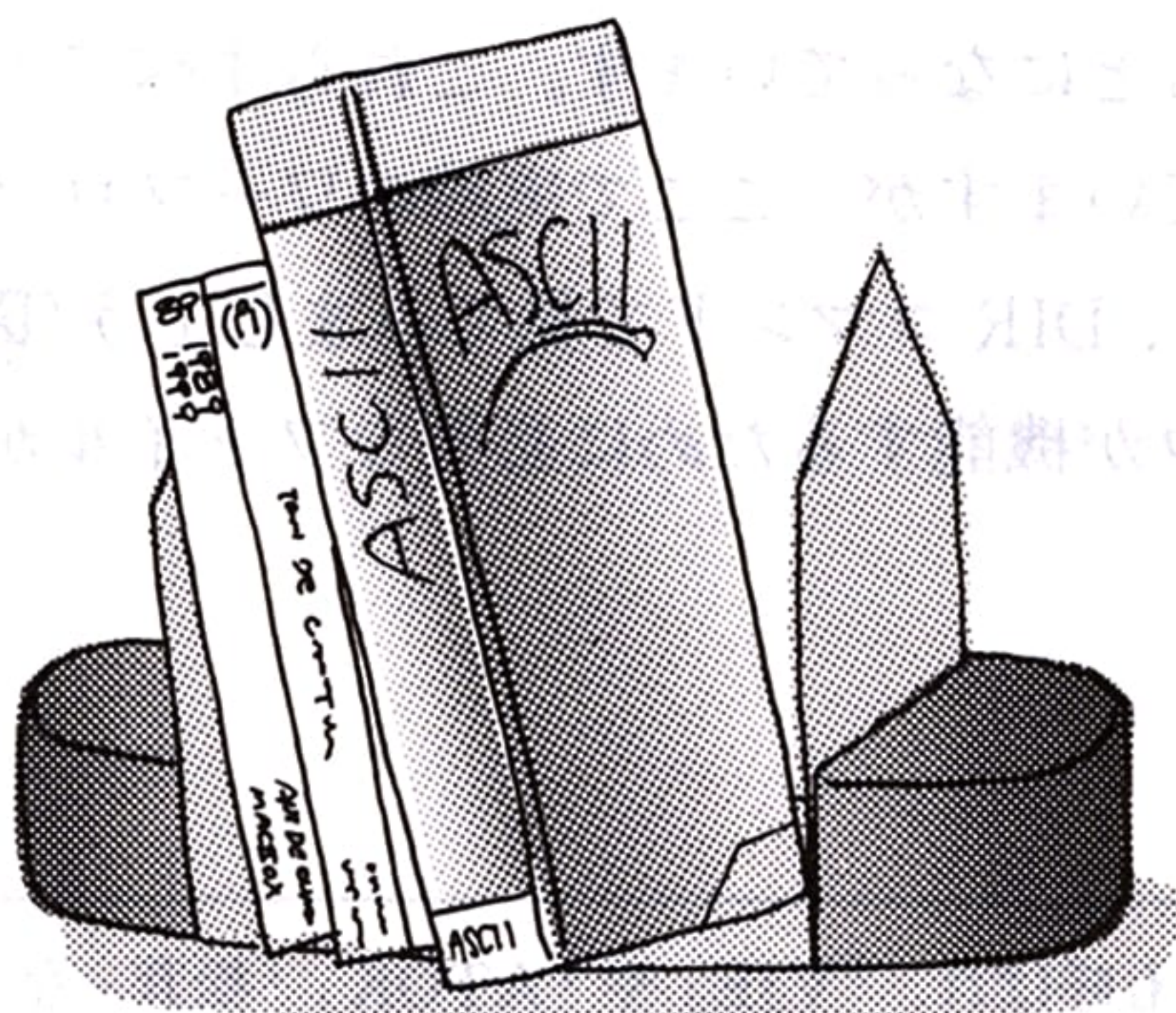
⑤文書が完成すると、作成された文書は「文書ファイル」としてディスク上に保存される

このような具合に、何をするにもすべてはファイルがもとになっています。これらすべてのファイルは、「ファイル名」を付けられてディスクに書き込まれていますが、ここで実際のワープロソフト一太郎を例に、そのシステムディスク上の各種のファイルを、DIRコマンドで見てみましょう(図6.1)。このように、ワープロのシステムディスクには、ワープロが機能するために必要なファイルが、それぞれのファイル名で書き込まれています。

---

* 日本語入力システムは、「フロントエンド・プロセッサ：FEP」とも呼ばれ、「VJE」や「ATOK」「松茸」などの、いくつかのシステムが使われている

```
A>DIR ⏎ ……………一太郎の常用ディスクの内容を見る

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    17190  83-09-19   16:20 ……………MS-DOSシステムのコマンド・プロセッサ
ATOK     DIC   454144  87-07-24   13:37 ……………一太郎の辞書ファイル
JXW      EXE   514264  87-06-15   12:00 ┐
JXW      OVL    78416  87-06-15   12:00 ├ 一太郎の本体のプログラムファイル
JXW      COM      942  87-06-15   12:00 ┘
ATOK6A   SYS    54265  87-06-15   12:00 ┐ 日本語入力システムのプログラムファイル
ATOK6B   SYS    21372  87-06-15   12:00 ┘
GAIJ             9392  87-06-15   12:00 ……………外字作成用プログラムファイル
PSET     PRD     8294  87-06-15   12:00 ……………一太郎用のプリンタ設定ファイル
FORMAT   COM    14742  85-05-19   13:59 ……………ディスクフォーマット・プログラムファイル
CONFIG   SYS       89  87-06-15   12:00 ……………MS-DOSのシステム構築用ファイル
AUTOEXEC BAT       35  87-06-15   12:00 ……………自動スタート・バッチファイル
JXWSYS           1550  89-07-20   19:05 ……………一太郎の作業環境定義ファイル
      14 個のファイルがあります.
    2048 バイトが使用可能です.

A>
```

図 6.1　一太郎の常用ディスク上の各種のファイル

6.2 ファイル名

DIR コマンドは、ディスク上のファイルの**ファイル名**や、そのファイルに関する各種の情報を表示するコマンドです。このコマンドは、今までに何度も実行例が示されていますので、みなさんは実際のファイル名がどのようなものなのか、その様子はすでにご承知とは思いますが、ここでその詳細について解説しましょう。

ディスク上のすべてのファイルは、ファイル名で識別されます。コンピュータは、このファイル名を唯一の頼りに、ディスク上の目的のファイルを捜し出し、いろいろな処理を行うわけです。

ファイル名は、次のように構成されています。

主ファイル名 (8文字まで) □□□□□□□□	ファイルタイプ (3文字まで) .□□□
絶対に必要	なくてもよい

このようにファイル名は、**主ファイル名**、**ピリオド**と、ファイルの種類(タイプ)を表すための**ファイルタイプ**の3つの部分から構成されています。ただし、この「ピリオド」は、上に示したようにファイルタイプの一部として考えてください。

ファイル名として、主ファイル名は絶対に必要ですが、ファイルタイプは特別なファイル(後述)以外は、付けなくてもかまいません。

ファイル名に使用できる文字には、次のような制限があります。

- 「A」~「Z」の英大文字/小文字(ただし、大文字/小文字は識別されない。小文字は大文字に変換される)
- 「0」~「9」の数字
- ! # $ % & ' ( ) - ^ { } @ ~ _ ` の各記号
- カタカナ
- 日本語入力システムによる漢字、ひらがな、カタカナ、各種記号など

ただし、日本語入力システムにより、漢字などの文字(いわゆる全角文字)を入力した場合は、1文字が、MS-DOS上での「通常」の文字や記号(いわゆる半角文字)の2文字分に相当します。つまりその場合、主ファイル名には「4文字」、ファイルタイプには「1文字+通常入力の1文字」しか使えないことになりますので注意してください(通常入力による半角文字の1文字は1バイト、日本語入力システムによる全角文字の1文字は2バイトで表されるため)。

またファイル名には、使ってはならない特別な名前が4つあります。本章の冒頭に「周辺装置もファイルである」と述べましたが、その周辺装置を表すための「ファイル名」として使われている名前、「CON」とか「PRN」などです(詳しくは9章を参照)。

以上の事項をまとめたものを**表6.1**に示します。

基本書式	□□□□□□□□	.□□□
名　称	・主ファイル名 ・「ファイル名」といえばここを指す場合もある	・ファイルタイプ ・拡張子 ・エクステンション
意　味	個々のファイルを表す主ネーム	ファイルの種類などを表す副ネーム
文字数	8文字まで	3文字まで (あるいは付けなくてもよい)
使用可能文字	「A」〜「Z」の英大文字/小文字（ただし、小文字は大文字に変換される） 「0」〜「9」の数字 「ア」〜「ン」のカナ ! # $ % & ' ( ) - ^ { } @ ～ _ ` の各記号 日本語入力システムによる漢字、ひらがな、カタカナ、各種記号　(注*)	
使用禁止ファイル名	「AUX」「CON」「PRN」「NUL」および各種の内蔵コマンドと同じ名前　(注*)	
*注意事項	・日本語入力システムにより入力した漢字などの文字は、通常の入力による文字の2文字分に当たる ・ファイル名にはどこにもスペースを入れてはいけない ・使用禁止ファイル名に、ファイルタイプを付けても使うことはできない 全「AUX.123」「CON.XYZ」「COPY.COM」なども使用不可	

表6.1　ファイル名についてのまとめ

では次に、具体的なファイル名の例をいくつか示しましょう。次のものは、いずれも正しいファイル名です。まず通常の入力方法(半角文字)による例です。

```
       A
       2
     A-5
DATA#123
       A.X
       B.X1
     A7B.XYZ
ABC(ｲﾛﾊ).A70
```

次に日本語入力システムによる文字(全角文字)と、通常入力による文字(半角文字)とを併用した例を示します。表示される文字の表面的な文字数と、内部的な文字数とは、半角／全角によって異なりますので注意が必要です。

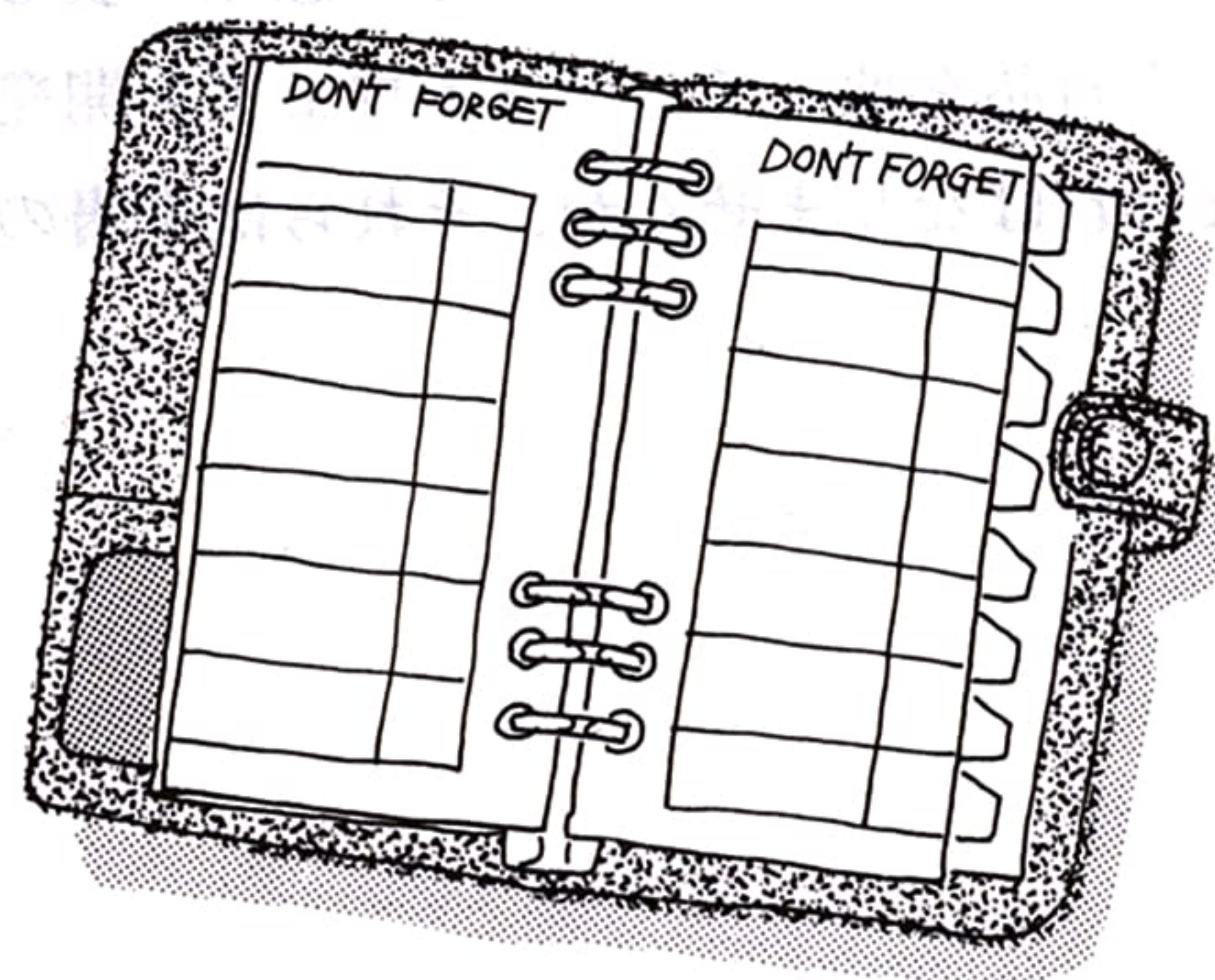
(半角文字／全角文字を併用した場合)

成績	……………ファイルタイプのないもの
３年Ｂ組.2期	……………2期の2は半角文字。これを全角文字で入力すると計4文字となり字数オーバーとなる
村瀬康治.Ｘ	……………Ｘは全角文字で入力しても2文字分なのでよい
〒番号.表	……………特殊記号を使ったもの
うりあげ.DAT	……………DATは半角文字

このようなものがMS-DOSのファイル名として自由に使用できます。

さて、ファイル名そのものについては、だいたい理解できたと思いますが、ファイル名に関連しては、まだ次の重要なことが残っています。

- ●ワイルドカード(ファイルマッチ)
- ●ディスクドライブの指定
- ●パス名

この3つです。これらは次の章で(パス名については11章で)実習を行いながら解説しますが、ここではいちおう、その言葉を頭に入れておいてください。

6.3 ファイルの種類 －特別の意味を持つファイルタイプ－

ファイル名のファイルタイプには、MS-DOSにとって特別な意味を持つ名称がいくつかあります。つまり、特定の種類のファイルには、ある決ったファイルタイプ名が付けられるのです。

具体例として、図6.1に示したワープロソフトの各ファイルを、ファイルタイプ別に並べ換えてみましょう(**表6.2**)。

ワープロソフトを構成する各ファイル	ファイルタイプ	(注)
COMMAND.COM FORMAT.COM JXW.COM	.COM	コマンドファイル(COMmand)
JXW.EXE	.EXE	エグゼキュータブルファイル (EXEcutable)
AUTOEXEC.BAT	.BAT	バッチファイル(BATch)
ATOK.DIC JXW.OVL PSET.PRD ATOK6A.SYS ATOK6B.SYS CONFIG.SYS	各種	ソフトメーカーが独自に付けたもので、MS-DOSで決められた特別な意味はない
GAIJ JXWSYS	——	ファイルタイプが付いてない例

表6.2　ワープロソフトの各ファイルをファイルタイプ別に分類した例(一太郎の場合)

ファイルタイプ別に分類すると**表6.2**のようになりますが、この中で「.COM」「.EXE」「.BAT」の3つのファイルタイプに注目してください。これら3つのファイルは、私たちがふだん最もよくお目にかかるものですので、とりあえずその意味を**表6.3**で説明しておきましょう。アセンブラや各種のプログラミング言語を使って、ソフトウェアを開発する人は、さらにいくつかの特定のファイルタイプを意識しなければなりませんが、それらは本書の続巻で解説します。

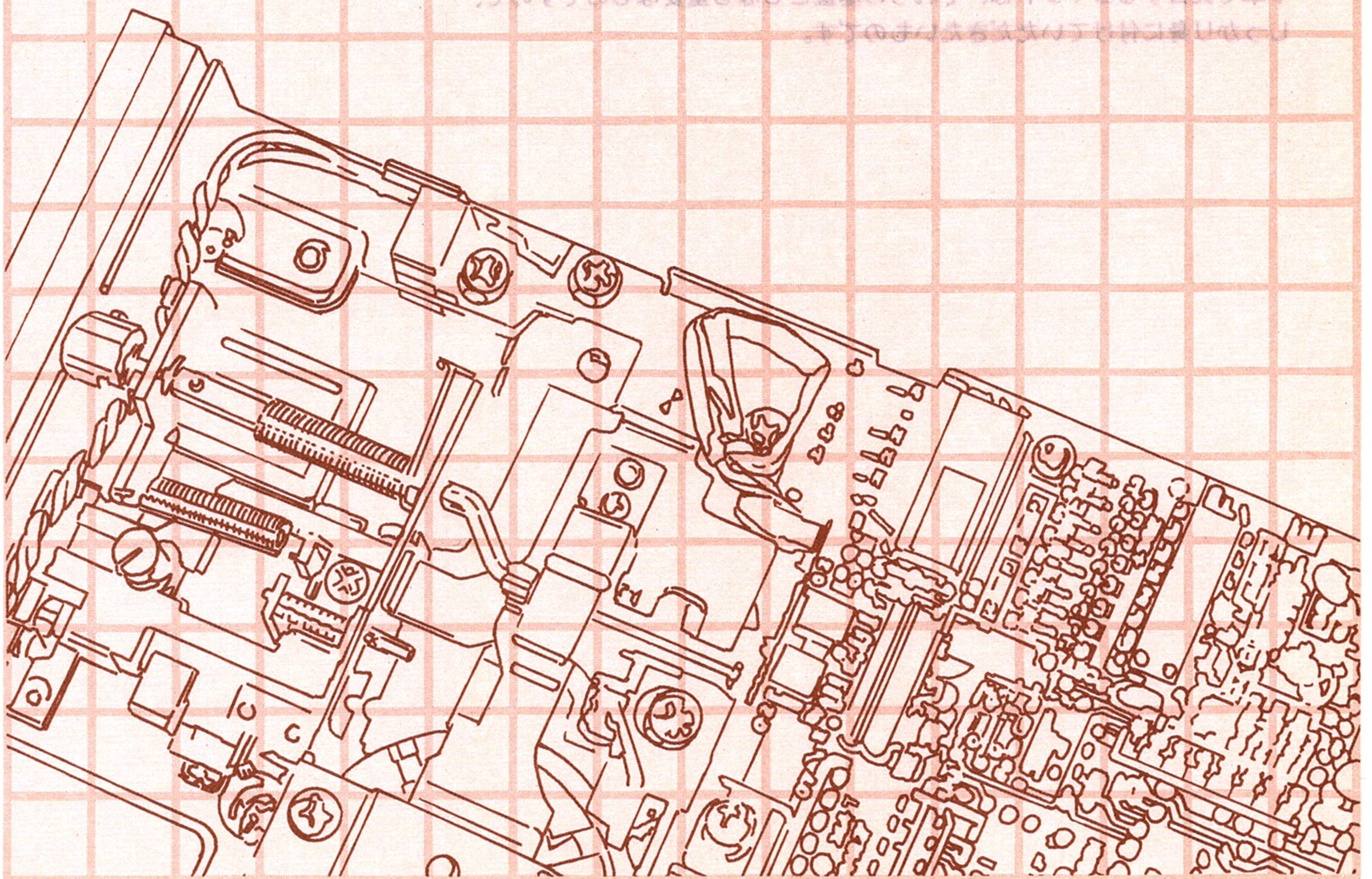
ファイルタイプ	これらが付いたファイルの意味
.COM	COMmand ファイルであることを示す。 この形式のものは、実行可能なマシン語のプログラムファイルで、大きさが 64K バイト以下の小規模のプログラムの場合に使われる。 このファイルタイプが付いたものは、主ファイル名をキー入力することにより、そのプログラムが実行される。 実行時には、このファイルのマシン語が、メモリの「所定の領域」に読み込まれ、実行される
.EXE	EXEcutable ファイルであることを示す。 上欄の「.COM」と同じ実行可能なマシン語のプログラムファイルである。大きさが 64K バイト以上のプログラムは、必ずこの形式となる。 実行時には、このファイルのマシン語が、「プログラムの冒頭で指定されているメモリ領域」に読み込まれ、実行される
.BAT	BATch ファイルであることを示す。 このバッチファイルについては、10 章で解説する。 バッチファイルの中でも、AUTOEXEC.BAT は特別なファイルであり、MS-DOS の起動時に、このファイルの内容で指示されたものが自動的に実行される

表 6.3　MS-DOS で決められている特別なファイルタイプとその意味

ファイルタイプには、このような特別の意味を持つものがあります。MS-DOS は、そのコマンドレベルにおいて、何らかのコマンドがキー入力されると、まずそれが DIR や TYPE コマンドのような内蔵コマンド(実行するためのプログラムが MS-DOS に内蔵されているコマンド)であるかどうかをチェックします。もし内蔵コマンドでなければ、キー入力されたコマンドと同じ「主ファイル名」を持つファイルがディスク上に存在するかをチェックし、存在していればその「ファイルタイプ」を調べ、それが上記 3 つのいずれかであれば、そのプログラムの実行を開始します。

このことから、実行可能なプログラムファイルの名前に、内蔵コマンドと同じファイル名は使えませんので、注意してください(作成しても、実行できない)。

さて次の章からは、ここまでの知識をもとに、MS-DOS の多くの機能の中でも日常的によく使うコマンドを、どんどん実習していきましょう。

7章
重要コマンドとその機能

さて本章では、日常よく使う基本的なコマンドを、どんどん具体的に実習していきます。ここまで読み進んできた知識をもとに、キーボードに向っていろいろなことを実際に試してみましょう。本章での実行例には、各コマンドの比較的単純な使い方を取り上げ、複雑な使い方は避けていますので、実習は楽に行うことができるでしょう。

なお、本章で登場する用語で、「内蔵コマンド」と「外部コマンド」がありますが、これについて詳しくは、次の8章で解説します。ここでの実習に際して、新しい用語などの不明部分がありましたら、随時先に目を通してみてください。6章、7章、8章は、どの章が先というのではなく、互いに関連があります。

まず最初は、DIRコマンドから実習していきますが、

ディスクをフォーマット処理する	FORMATコマンド
ディスクをまるごとコピーする	DISKCOPYコマンド
MS-DOSシステムをコピーする	SYSやFORMAT/Sコマンド

などは、すでに3章で実習解説していますので、ここでは取り上げていません。また、「階層ディレクトリ」関係のコマンドは、11章で登場します。本章で実習するコマンドは、それらの基盤ともなる重要なものですので、しっかり身に付けていただきたいものです。

7.1 ファイル情報の表示 DIR(ディアイアール)

─ 内蔵コマンド、ワイルドカード使用可 ─

DIR とは DIRectory のことであり、たとえば電話帳を「telephone directory」というような具合に、「… 帳」「… 録」「… 簿」といったような意味です。

DIR コマンドは、ファイルのディレクトリ ── つまり、ディスク上のファイルの、いわば「登録名簿」に記載されている各種の情報を表示します。

この登録名簿は、物理的にはそれぞれのディスクの最も外周部に位置する**ディレクトリ領域**に書き込まれています(ディレクトリ領域については 4.2 節で触れている)。DIR コマンドを実行すると、MS-DOS は、この領域のデータを読み出してその内容を表示するわけです。

DIR コマンドを実行するための書式を次に示します。「*x*：」はドライブ名で、カレントドライブ(現在「メイン」となっているディスクドライブ)の場合は省略できます(カレントドライブ、ドライブ名、ワイルドカードなどの解説は、このあとすぐに登場する)。

```
A>DIR x：⏎
A>DIR x：ファイル名⏎
A>DIR x：ワイルドカード⏎
```

DIR コマンドには、さらに次のスイッチ(コマンドオプションのこと)による付加機能があります。いずれも上記の書式の最後部(⏎の前)に付け加えます。「/」の前にスペースを入れても入れなくても、どちらでもかまいません。

/P ……… ページ表示(1 画面ごとに区切を置いて表示する)
/W ……… ワイド表示(ファイル名のみを 1 行に 5 個ずつ表示する)

またこの 2 つのスイッチを、「/P/W」のように同時に指定することもできます。

では、ドライブ A：に実習のためのディスクとして、市販の MS-DOS システムディスクのコピーをセットして、DIR コマンドを実行してみましょう(図 7.1)。ワイルドカードを使う実行例は、次節で実習します。

DIR コマンドは内蔵コマンドであり、実行するためのプログラムは MS-DOS に内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。

```
A>DIR ↵
```
……本来は「DIR A:↵」であるが、カレントドライブがA:なので(プロンプトがA>)、このA:は省略してある。ドライブA:上のすべてのファイルの情報が表示される

```
ドライブ A: のディスクのボリュームラベルはありません.
ディレクトリは A:¥
```
ファイル名 / ファイルの大きさ(バイト) / 作成年-月-日 / 作成時刻

```
COMMAND  COM    24931  88-07-13   0:00
CONFIG   SYS       87  89-07-29  17:25
ATOK6A   SYS    54271  87-06-15  12:00
ATOK6B   SYS    21372  87-06-15  12:00
ATOK     DIC   454144  89-07-30  16:50
  ～～～～～～～～～～～～～～～～～～～～
TREE     EXE     9608  88-07-13   0:00
ATTRIB   EXE     9154  88-07-13   0:00
EDLIN    EXE     7922  88-07-13   0:00
README   DOC    10578  88-07-13   0:00
AUTOEXEC BAT        7  89-07-30  13:33
      21 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```
357376 バイトが使用可能です.……カレントドライブの使用可能な残りディスク容量

```
A>DIR/P ↵
```
……ページ表示のスイッチ「/P」を付けて実行する。これも本来は「DIR A:/P↵」である

```
ドライブ A: のディスクのボリュームラベルはありません.
ディレクトリは A:¥

COMMAND  COM    24931  88-07-13   0:00
CONFIG   SYS       87  89-07-29  17:25
ATOK6A   SYS    54271  87-06-15  12:00
ATOK6B   SYS    21372  87-06-15  12:00
ATOK     DIC   454144  89-07-30  16:50
PRINT    SYS     5855  88-07-13   0:00
RSDRV    SYS     7352  88-07-13   0:00
  ～～～～～～～～～～～～～～～～～～～～
SORT     EXE     2138  88-07-13   0:00
TREE     EXE     9608  88-07-13   0:00
ATTRIB   EXE     9154  88-07-13   0:00
EDLIN    EXE     7922  88-07-13   0:00
README   DOC    10578  88-07-13   0:00
AUTOEXEC BAT        7  89-07-30  13:33
      21 個のファイルがあります.
   357376 バイトが使用可能です.
準備ができたらどれかキーを押してください. x
```
1ページ目の表示画面がいっぱいになると、このように表示が一時ストップする

準備ができたらどれかキーを押してください.……ここで表示が一時ストップしているので、適当なキーを入力すると次のページが表示される

```
A>
```
2ページ目の表示

(このディスク上のファイルの表示は、ちょうど1ページにおさまった)

— 図 7.1 —(次ページに続く)

```
A>DIR/W ↵
```
……………ワイド表示のスイッチ「/W」を付けて実行する。これも本来は「DIR A:/W ↵」であるがA:は省略してある

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    CONFIG   SYS    ATOK6A   SYS    ATOK6B   SYS    ATOK     DIC
PRINT    SYS    RSDRV    SYS    MOUSE    SYS    FORMAT   EXE    DISKCOPY EXE
CHKDSK   EXE    SYS      EXE    FC       EXE    FIND     EXE    MORE     COM
SORT     EXE    TREE     EXE    ATTRIB   EXE    EDLIN    EXE    README   DOC
AUTOEXEC BAT
        21 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```
ファイルの大きさや作成日／時の表示が省略され、ファイル名のみが1行に5個ずつ表示されている

```
A>DIR FORMAT.EXE ↵
```
……………本来は「DIR A:FORMAT.EXE ↵」
カレントドライブ上(A:)のファイル「FORMAT.EXE」を調べる

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

FORMAT   EXE    97766  88-07-13    0:00
         1 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```

```
A>DIR ABCD.XYZ ↵
```
……………指定したファイルがカレントドライブ上(A:)に存在しない場合

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

ファイルが見つかりません.

A>
```

図 7.1　DIR コマンドの実行例

7.2 ワイルドカード記号（ファイルマッチ記号）

ファイル名については、6.2 節で詳しく解説しましたが、その中では触れなかった、たいへん重要なファイル名の使い方があります。これは、**ワイルドカード**あるいは**ファイルマッチ**と呼ばれ、ファイル名の文字列構成に、何らかの共通項があるファイルのグループを、まとめて表したり、指定したりすることができる機能です。つまりワイルドカード記号は、トランプのジョーカーと同様に、何らかの文字や文字列の代用として使うことができるのです。またワイルドカードは、「合う」とか「合致する」とかいったような意味で「ファイルマッチ」とも呼ばれますが、本書では、ワイルドカードと呼ぶことにしましょう。

ワイルドカードには、「?」および「*」記号を使います。それらの記号は、

? ……… すべての文字１文字の代用（主ファイル名やファイルタイプの、それぞれの最後の文字に使った場合は、そこに実際の文字がなくてもマッチする）

* ……… すべての文字列の代用（字数がゼロ（文字がない場合）にもマッチする）

という意味があり、たとえば、「?」記号を使った場合、

AB?DE.XYZ　は　AB□DE.XYZ

というすべてのファイル名に相当します（１つの□は、すべての１文字に相当する）。□の中には、ファイル名に使用できるどのような１文字がはいっていてもよいわけです。具体例としては、

ABCDE.XYZ
AB5DE.XYZ
ABアDE.XYZ

などにマッチします。また「?」記号は、１つのファイル名のどこにでも使うことができ、いくつ使ってもかまいません。たとえば、

AB??E.X??　は　{ AB□□E.X□□ / AB□□E.X□ / AB□□E.X }

（「?」記号は、主ファイル名あるいはファイルタイプの最後の文字に使った場合に限り、実際の文字がなくてもマッチする）

に相当するすべてのファイル名にマッチします。

次に、「*」記号の使い方を示しましょう。たとえば、

*.XYZ　は
- □.XYZ
- □□.XYZ
- □□□.XYZ
- ∫
- □□□□□□□□.XYZ

などにマッチします。つまり、「.」の左側の主ファイル名が何であってもよいわけです。同様に、

ABCDE.*　は
- ABCDE
- ABCDE.□
- ABCDE.□□
- ABCDE.□□□

などにマッチします。つまり「.」の右側のファイルタイプが何であっても、あるいは、ファイルタイプがなくても(「文字がない場合」に相当する)よいわけです。さらに、「?」と「*」とを組み合わせて使うこともできます。たとえば、

ABC*.X?Z　は
- ABC.X□Z
- ABC□.X□Z
- ABC□□.X□Z
- ∫
- ABC□□□□□.X□Z

などにマッチします。

では実際に、これらの実行例を示しましょう(**図 7.2**)。ここでの実習用に使用するディスクは、前節と同じシステムディスクのコピーです。

```
A>DIR *.COM ⏎

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM     24931  88-07-13     0:00
MORE     COM       303  88-07-13     0:00
        2 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```

……本来は「DIR A:*.COM⏎」。カレントドライブ上(A:)のファイルタイプが「.COM」であるすべてのファイルを調べる

図7.1に表示されているファイルの中から、ファイルタイプが「.COM」のものだけが表示されている

— 図 7.2 — (次ページ以下に続く)

```
A>DIR *.DOC ↵ ……………ファイルタイプが「.DOC」であるすべてのファイルを調べる

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

README   DOC    10578  88-07-13    0:00 ……………マッチするものが1個あった
         1 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```

```
A>DIR C* ↵ ……………ファイル名の最初の文字が「C」であるすべてのファイルを調べる

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13    0:00 ┐
CONFIG   SYS       87  89-07-29   17:25 ┝ マッチするものがこれだけあった
CHKDSK   EXE    10384  88-07-13    0:00 ┘
         3 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```

```
A>DIR F*.?X? ↵ ……………ファイル名の最初の文字が「F」で、かつファイルタイプの2番目の文字
                        が「X」であるすべてのファイルを調べる
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

FORMAT   EXE    97766  88-07-13    0:00 ┐
FC       EXE    15302  88-07-13    0:00 ┝ マッチするものがこれだけあった
FIND     EXE     6573  88-07-13    0:00 ┘
         3 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```

```
A>DIR S*.E*
```
……主ファイル名の最初の文字が「S」で、ファイルタイプの最初の文字が「E」であるすべてのファイルを調べる

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

SYS      EXE    25480  88-07-13    0:00
SORT     EXE     2138  88-07-13    0:00
        2 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```
マッチするものがこれだけあった

```
A>DIR ???.EXE
```
……主ファイル名が3文字以下で、ファイルタイプが「.EXE」であるすべてのファイルを調べる

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

SYS      EXE    25480  88-07-13    0:00
FC       EXE    15302  88-07-13    0:00
        2 個のファイルがあります.
   357376 バイトが使用可能です.
A>
```
マッチするものが2個あった

このような「*」の使い方には要注意

```
A>DIR *ND.COM
```
……主ファイル名の最後の2文字が「ND」で、ファイルタイプが「COM」であるファイルを調べるつもり

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13    0:00
MORE     COM      303  88-07-13    0:00
        2 個のファイルがあります.
   357376 バイトが使用可能です.

A>
```
……ところがこのようなファイルまでマッチしている

このように主ファイル名、ファイルタイプのそれぞれにおいて、「*」記号の後ろの文字の指定は無視されるので注意が必要である

図 7.2　ワイルドカード記号(ファイルマッチ)の使用例

7.3 ドライブA：、B：とカレントドライブ

パーソナル・コンピュータをいろいろな仕事に活用する場合、接続されているディスクドライブに関しては、

フロッピーディスクのみの場合 …… 2ドライブ以上
ハードディスクの場合 ……………… 1台のハードディスクドライブ+1ドライブ以上（できれば2ドライブ以上）のフロッピーディスクドライブ

を持つシステムでなければ、実用にはなりません。フロッピーディスクは、2ドライブで1つのディスク装置ぐらいに考えてください。また、ハードディスクを持つシステムは、ファイルのバックアップの意味からも、必ずフロッピーディスクを付けて、重要なデータは、フロッピーディスクにコピーして、バックアップとして保管するようにしましょう。

このように、どのようなディスク装置のシステムでも、普通は複数のドライブを持つことになります。本節ではこれら複数のドライブを、どのように使えばよいのか、その基本的な取り扱いについて実習解説します。

ではまず、ディスクドライブの呼び方から始めましょう。ディスクドライブの前面パネルには、「ドライブ1」「ドライブ2」などと書かれている場合が多いようですが（各メーカーによって異なる）、MS-DOSでは2.2節で述べたように、各ドライブを順に、

ドライブ　A：
ドライブ　B：
ドライブ　C：
　　∫

と呼びます。接続できるドライブの台数は、MS-DOSのバージョン3.3では理論的には26台まで可能ですが、まあ常識的には、多い場合でもドライブA：～F：の6台程度のシステムでしょうか（たとえば、ハードディスク2台、5インチおよび3.5インチ2HDのフロッピーディスクドライブが2台ずつ、計6台程度のシステムは十分に考えられる）。

本章で実習解説を行うためのコンピュータシステムのディスクドライブ構成を、**図7.3**に示しておきます。ただしここでの実習解説は、あくまで1つの例にすぎませんので、それぞれの実行例の意味を、自分のシステムに当てはめながら理解するようにしてください。

ここでの通常の運用は、5インチ側をドライブA:、B:としている
5インチ2HDドライブ
ドライブA:
ドライブB:
ドライブC:　ドライブD:
3.5インチ2HDドライブ

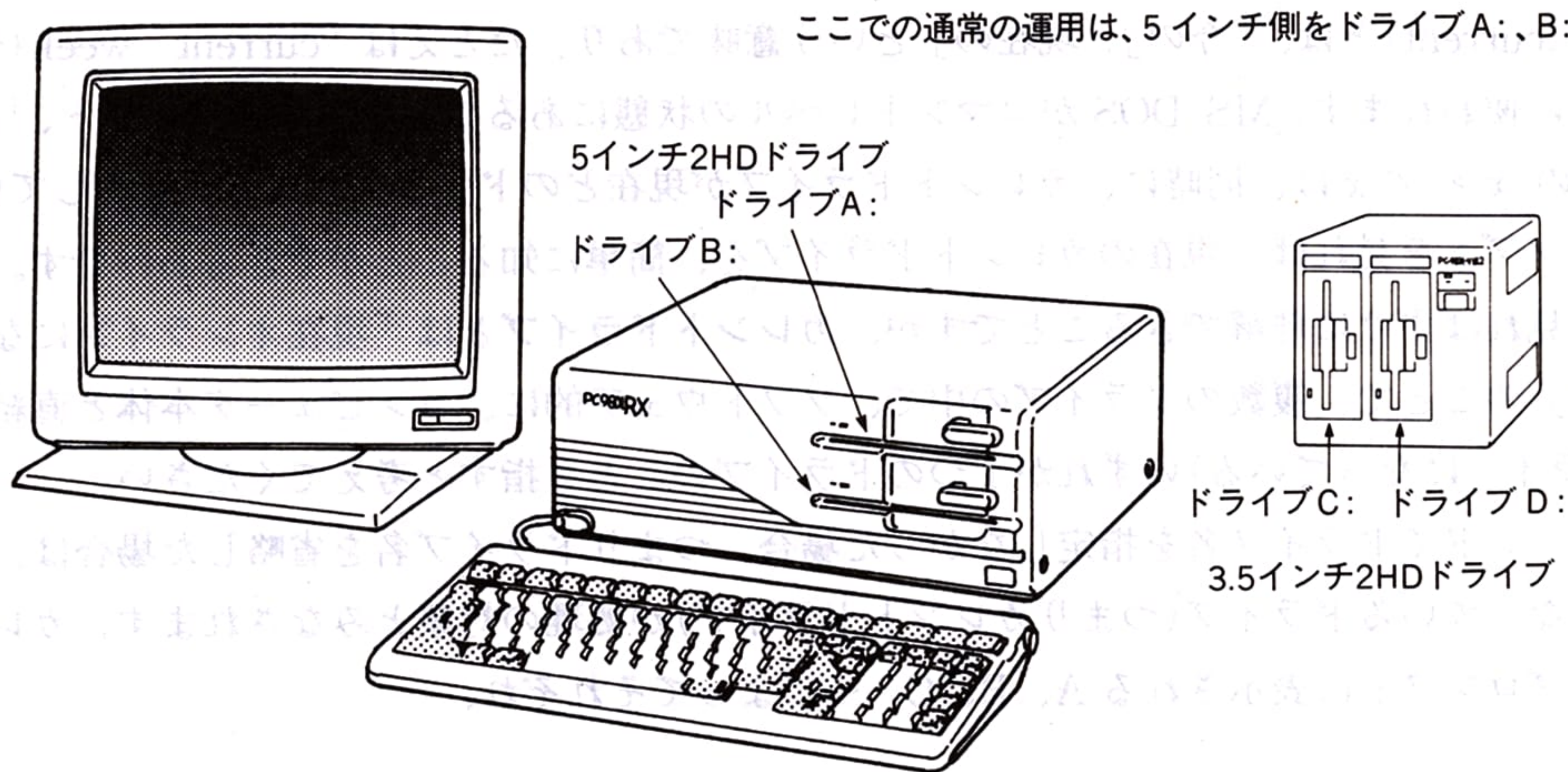



図 7.3　ここでの実習で想定しているマシンのディスクシステムの構成

このシステムには、5インチ2HDのドライブのほかに、ラップトップマシンのディスクとのデータ交換などの目的で、3.5インチ2HDのドライブが接続されています。

さて、ここでの実習は、各ドライブのドライブ名を、

5インチドライブ側を ……… **ドライブA：、B：**
3.5インチドライブ側を …… **ドライブC：、D：**

として話を進めましょう(このようなドライブ名の順序は、5インチのシステムディスクから立ち上げればこのような設定になり、3.5インチのシステムディスクから立ち上げれば3.5インチ側がドライブA：、B：となるなど、起動時の設定条件によって決定される)。

また、それぞれのドライブには、次のような内容のディスクがセットされているものとします(現実的ではないが)。

ドライブA： …… MS-DOSの常用システムディスク
ドライブB： …… Multiplanのディスク
ドライブC： …… ワープロ(一太郎)のディスク
ドライブD： …… ワープロで作成される文書の保存用ディスク

では、ディスクドライブに関するいろいろな操作を、実行例を示しながら実習していきましょう。同じコマンドでも、ドライブ名の指定や、カレントドライブのチェンジにより、処理の対象となるディスクが異なってくることに注目してください。

■ カレントドライブのチェンジ

カレント(current)とは、「今の」「現在の」という意味であり、たとえば「current　week(今週)」というように使われます。MS-DOSがコマンドレベルの状態にあることを示すプロンプト、「A>」や「B>」などの「x>」のxは、同時に、カレントドライブが現在どのドライブであるかを示しています。つまりプロンプトを見れば、現在のカレントドライブを、簡単に知ることができるわけです。

実行例を見ればすぐに理解できることですが、**カレントドライブ**とは「現在オンラインになっているドライブ」のことで、複数のドライブの中で、ソフトウェア的に、コンピュータ本体と直結されている(オンラインになっている)いずれか1つのドライブのことを指すと考えてください。

各種のコマンドでドライブ名を指定しなかった場合、つまりドライブ名を省略した場合は、現在オンラインになっているドライブ(つまりカレントドライブ)が処理の対象とみなされます。カレントドライブは、プロンプトに表示されるA、B、C、…によってそれぞれ、

A> …… ドライブA：がカレントドライブ
B> …… ドライブB：がカレントドライブ
C> …… ドライブC：がカレントドライブ

となります。

カレントドライブを任意のドライブに変更するのは非常に簡単であり、

```
A>x:⏎
```

と入力します。その時点でプロンプトは「x>」に変わり、カレントドライブがドライブx：に移ります(図7.4)。

カレントドライブとは、「現在自分のもの」になっているドライブと考えればよいでしょう。

■ 各種のコマンドやプログラムの実行におけるドライブの指定

MS-DOSの各種のコマンドや、各種のアプリケーションソフトを実行する際、実行するプログラムや対象となるファイルがカレントドライブ上にあれば問題なく実行できます。もし、目的のプログラムやファイルが別のドライブ上にある場合は、そのドライブがカレントドライブになるように、カレントドライブを変更(チェンジ)した上で実行すればよいわけです。しかしこのような方法では、必要とするファイルが、すべてカレントドライブ上になくてはならないことになります。これでは制約が多すぎて能率的な仕事はできません。

そこで、カレントドライブには関係なく、目的のファイルのファイル名の頭に、そのファイルが存在するドライブ名を付ければ、カレントドライブでなくても任意のドライブ上のファイルを操作することができるようになっています。

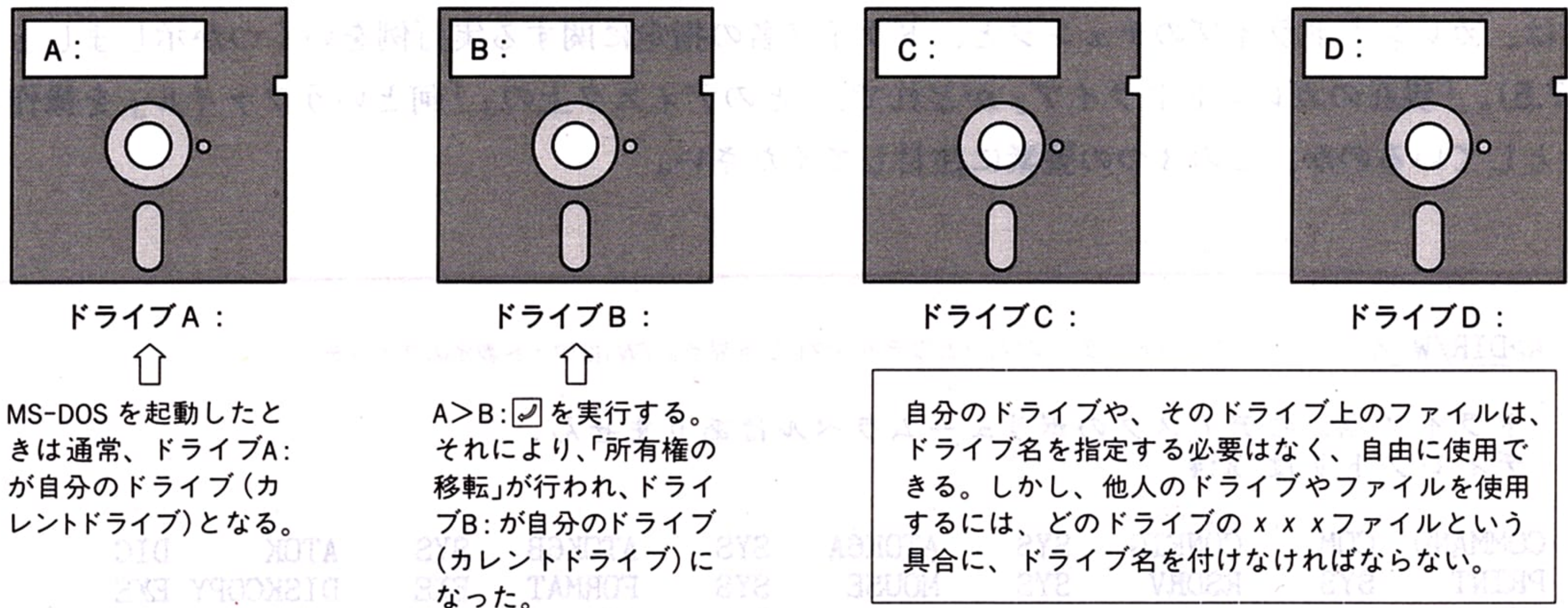



図7.4　カレントドライブと、そのチェンジの考え方

具体的には、たとえば目的のファイル名を「ABCD.XYZ」とし、そのファイルが存在するディスクドライブをA：、B：、C：、…とすると、

ドライブA：上のファイル「ABCD.XYZ」を指定するには …… A：ABCD.XYZ
ドライブB：上のファイル「ABCD.XYZ」を指定するには …… B：ABCD.XYZ
ドライブC：上のファイル「ABCD.XYZ」を指定するには …… C：ABCD.XYZ

のように、ファイル名の先頭に、そのファイルが存在する「ドライブ名」と「：」を付けます。コロン「：」は、単独記号と考えず、ドライブ名の一部と考えた方がよいでしょう。このことから、本書ではドライブA、ドライブBを「ドライブA：」「ドライブB：」と記しているわけです。

また、DIRコマンドやCHKDSKコマンドなどで、ファイル名を指定しない場合は、

DIR　B：………… ドライブB：上のファイル情報を見る
CHKDSK　C：…… ドライブC：上のディスクをチェックする

のようにドライブ名だけを指定します。

なお、以下の実行例にはCHKDSKコマンドが登場しますが、このコマンドについては、7.9節で解説します。CHKDSKコマンドを実行するには、ディスク上にそのプログラムファイル「CHKDSK.EXE」（あるいは.COM）が必要ですので確認しておいてください。つまり、CHKDSKコマンドは、そのプログラムがMS-DOS内部に組み込まれている「内蔵コマンド」ではなく、ディスク上に存在する「外部コマンド」です。また、TYPEコマンドも登場しますが、このコマンドは内蔵コマンドであ

り、そのプログラムはMS-DOSに内蔵されていますので、ディスク上に存在する必要はありません(TYPEコマンドは7.4節、内蔵コマンド、外部コマンドについては8章で解説する)。

では、カレントドライブのチェンジと、ドライブ名の指定に関する実行例をいくつか示しましょう(図7.5)。「現在のカレントドライブ」がどれで、「どのディスク上の」「何というファイル」を操作しようとしているのか、この3つの要素に注目してください。

```
A>DIR/W ⏎ ……………カレントドライブ(A:)上のディレクトリを見る。「/W」はワイド表示のスイッチ

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    CONFIG   SYS    ATOK6A   SYS    ATOK6B   SYS    ATOK     DIC
PRINT    SYS    RSDRV    SYS    MOUSE    SYS    FORMAT   EXE    DISKCOPY EXE
CHKDSK   EXE    SYS      EXE    FC       EXE    FIND     EXE    MORE     COM
SORT     EXE    TREE     EXE    ATTRIB   EXE    EDLIN    EXE    README   DOC
AUTOEXEC BAT
       21 個のファイルがあります.
    357376 バイトが使用可能です.          これは常用のMS-DOSのシステムディスク

A>DIR B:/W ⏎ ……………カレントドライブ(A:)からドライブB:上のディレクトリを見る

 ドライブ B: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは B:¥

COMMAND  COM    FORMAT   EXE    MP       COM    MP       COD    MP       SYS
MP       DAT    MP       HLP    MP       PIF    MOUSE    SYS    SETMP    INI
601_P    CPD    601_DL   CPD    601_DP   CPD    601_L    CPD    201H     CPD
201H_D   CPD    201H_DS  CPD    201H_S   CPD    ATOK6A   SYS    ATOK6B   SYS
ATOK     DIC    CONFIG   SYS    AUTOEXEC BAT    MP       INI
       24 個のファイルがあります.
     34816 バイトが使用可能です.          これはMultiplanのディスク

A>B: ⏎ ……………カレントドライブをA:からB:にチェンジする

B>DIR/W ⏎ ……………プロンプト「B>」に注目。カレントドライブ(B:)上のディレクトリを見る

 ドライブ B: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは B:¥

COMMAND  COM    FORMAT   EXE    MP       COM    MP       COD    MP       SYS
MP       DAT    MP       HLP    MP       PIF    MOUSE    SYS    SETMP    INI
601_P    CPD    601_DL   CPD    601_DP   CPD    601_L    CPD    201H     CPD
201H_D   CPD    201H_DS  CPD    201H_S   CPD    ATOK6A   SYS    ATOK6B   SYS
ATOK     DIC    CONFIG   SYS    AUTOEXEC BAT    MP       INI
       24 個のファイルがあります.
     34816 バイトが使用可能です.   結果は上の実行例と同じドライブB:上のMultiplanのディスク
```

— 図7.5 — (次ページ以下に続く)

```
B>CHKDSK ⏎
コマンドまたはファイル名が違います.
```
……カレントドライブ(B:)上の「CHKDSK」コマンドを実行するつもり
……ドライブB：にはCHKDSKのプログラムがない

```
B>A:CHKDSK ⏎
ディスク MULTIPLAN   は 1989-7-19 14:56 に作成されました.

  1250304 バイト : 全ディスク容量
    95232 バイト : 3 個のシステムファイル
  1120256 バイト : 24 個のユーザーファイル
    34816 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   454160 バイト : 使用可能メモリ
```
……カレントドライブ(B:)から、A:上のCHKDSKコマンドを実行し、カレントドライブ(B:)上のディスクをチェックする。本来は「B>A:CHKDSK B:⏎」である
B：側のディスクの状態が表示されている

```
B>A:⏎
```
……カレントドライブをA：に戻す

```
A>CHKDSK B:⏎
ディスク MULTIPLAN   は 1989-7-19 14:56 に作成されました.

  1250304 バイト : 全ディスク容量
    95232 バイト : 3 個のシステムファイル
  1120256 バイト : 24 個のユーザーファイル
    34816 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   454160 バイト : 使用可能メモリ
```
……カレントドライブ(A:)上のCHKDSKコマンドを実行し、ドライブB:のディスクをチェックする
当然ながら、上と同じ結果

```
A>MP⏎
コマンドまたはファイル名が違います.
```
……カレントドライブ(A:)上のMultiplanのプログラム「MP」を実行するつもり。本来は「A>A:MP⏎」
……ドライブA:上にMultiplanはない

```
A>B:⏎
```
……カレントドライブをB:にチェンジする

```
B>MP⏎
```
……再度実行する。Multiplanのプログラムは B:側に存在している

↓

```
#1      1        2        3        4        5        6        7
  1 ████████
  2
  3
  4
 17
 18
 19
コマンド: A/文字 B/空白 C/複写 D/削除 E/修正 F/書式 G/セル移動 H/解説 I/挿入 L/保護
  M/移動 N/名前 O/オプション P/印刷 Q/終了 R/実行 S/並替 T/転送 V/数式 W/窓 X/連結
コマンドを選択するか，英字を入力してください.
R1C1                       100% Free     Multiplan: TEMP
```
Multiplanが起動した

↓

Multiplanを終了して、カレントドライブはB:の状態

B>DIR C:/W ……………カレントドライブ(B:)から、ドライブC:に対してDIRコマンドを実行する

ドライブ C: のディスクのボリュームラベルはありません.
ディレクトリは C:¥

ドライブC:上のファイルが表示されている

```
COMMAND COM    ATOK    DIC    JXW      EXE    JXW     OVL    JXW    COM
CACHE   SYS    ATOK6A  SYS    ATOK6B   SYS    GAIJ           FORMAT COM
PSET    PRD    CONFIG  SYS    AUTOEXEC BAT    JXWSYS
       14 個のファイルがあります.
     2048 バイトが使用可能です.
```

このディスクは一太郎のシステムディスクである

B>C: ……………カレントドライブをC:に移す

C>JXW ……………一太郎を実行する。そのプログラムファイル名が「JXW.COM」

```
[1]                [2]               《 一 太 郎 》     1 頁   1 行   1 列
[1] ■-------10---------20---------30---------40---------50---------60---------70-+---
  |
  |
  |
  |
  |
  |
  |
-------------------------------------------------------------------------------------
A:文字入力                                               【挿入】【確定入力】
                                                              連ローマ字漢字
```

一太郎が立ち上がった

一太郎を終了して、MS-DOSのコマンドレベルに戻った

C>CHKDSK D: ……………CHKDSKコマンドを実行して、ドライブD:上のディスクをチェックするつもり
コマンドまたはファイル名が違います. ……………カレントドライブ(C:)上にはCHKDSKのプログラムがない

C>A:CHKDSK D: ……………カレントドライブ(C:)からA:上のCHKDSKコマンドを実行して、ドライブD:上のディスクをチェックする

```
 1250304 バイト : 全ディスク容量
  424960 バイト : 25 個のユーザーファイル
  825344 バイト : 使用可能ディスク容量

  655360 バイト : 全メモリ
  454160 バイト : 使用可能メモリ
```

ドライブA:上のCHKDSKコマンドが実行されて、ドライブD:のワープロの文書ディスクの状態が表示されている

```
C>B:⏎ ……………カレントドライブをB:に移す

B>TYPE README.DOC⏎ ……TYPEコマンドで、カレントドライブ(B:)上のファイル「README.DOC」を
                        ディスプレイに表示する。本来は「B>TYPE B:README.DOC⏎」である
ファイルが見つかりません．……表示対象のファイル「README.DOC」はカレントドライブ(B:)上にはなかった

B>TYPE A:README.DOC⏎ ……………今回は「README.DOC」の頭にドライブ指定「A:」を付けて実行する

1．　バックアップ作成時の注意
                       ドライブA:上のファイル「README.DOC」が表示されている

　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタ
フォーマット（７２０ＫＢ）でフォーマットされています。
ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
　　　　　　　　　　　　　　　　　/９スイッチを指定してください。
記録されるため、読み出しエラーが起
ります。
　このようなセクタは、「スキップセクタ」と呼び、システムでデータ記憶領域と
して使用しないようにして、お客様のデータが失われることのないよう、予防して
います。
　固定ディスクに対して、ＣＨＫＤＳＫコマンドを実行すると、「×××××バイ
ト：スキップセクタ」と表示される場合がありますが、これは、このような予防処
置がされていることを表しますので、「スキップセクタ」が表示されても安心して
ご使用いただけます。

　なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品
に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけ
ます。

B>A:⏎ ……………カレントドライブをA:に戻す

A> ……………カレントドライブはA:
```

図7.5　カレントドライブとドライブ名の指定例

ここで、カレントドライブの指定について整理しておきましょう。次のように考えてみてください。

> カレントドライブとは、自分のディスクのことと考える。要するに、カレントドライブ内のファイルは「自分のファイル」である。したがって、自分が自分のディスクやファイルを使うのに、他人に断る（ドライブ名を指定する）必要はない。
>
> カレントドライブ以外のディスクやファイルは他人のものであり、使用する場合には、「誰々の」というお断りのドライブ名を付けなければならない。

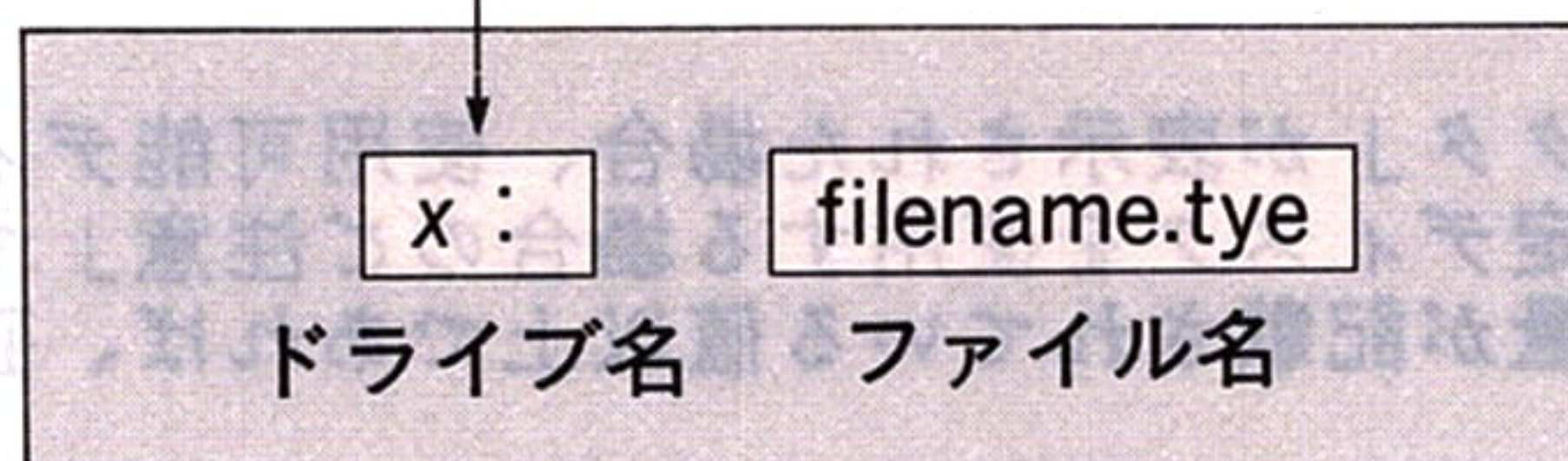
こう考えれば、各種のコマンドの書き方は、次の表のように解釈できます。カレントドライブ（プロンプトの「A>」「B>」など）に注目してみてください。

A>DIR ⏎ ⇩（本来は） A>DIR A:⏎	自分のディスク（A:）のファイル情報を表示せよ カレントドライブがA>なので、A:は省略可能
B>A:CHKDSK ⏎ ⇩（本来は） B>A:CHKDSK B:⏎	他人のドライブ(A:)上のCHKDSKコマンドを実行して、自分のディスク「(B:)」の状況を表示せよ B:は省略可能、A:は不可
B>A:CHKDSK A:⏎	他人のドライブ(A:)上のCHKDSKコマンドを実行し、同じ他人のディスク「(A:)」の状況を表示せよ 2つの「A:」は省略不可
A>CHKDSK B:⏎ ⇩（本来は） A>A:CHKDSK B:⏎	自分のディスク(A:)のCHKDSKプログラムを実行し、他人のディスク（B:）の状況を表示せよ A:は省略可能、B:は不可

CHKDSKコマンドのプログラムファイルがドライブA:上に存在しているとする

つまり、複数のドライブの中の、任意のファイルを指定するには、ファイル名の先頭にドライブ名を付ければよいわけです。結局、ファイルの完全な指定*は次のようになります。

カレントドライブの場合は省略できる

x :	filename.tye
ドライブ名	ファイル名

＜ファイルの完全な指定*＞

また、次のようにドライブ名のみを指定する場合もあります。

```
A>DIR  B:⏎
```

この場合も、指定するドライブがカレントドライブであれば省略することができます。

A>DIR　A:⏎ → A>DIR⏎

＊「完全な指定」は、厳密には11章で解説するディレクトリの指定も必要であるが、ここではまだ触れていない。

7.4 文字ファイルの内容を読む TYPE（タイプ）

― 内蔵コマンド、ワイルドカード使用不可 ―

ファイルの内容を、TYPE コマンドを使って表示(TYPEout：タイプアウト)することができます。つまり、目的のファイルの内容を読み出し、画面に表示するコマンドです。ただし、TYPE コマンドが表示できるファイルは**文字ファイル**に限られます。文字ファイルというのは、文字データだけで構成されているファイルのことで、**アスキーファイル**とか**テキストファイル**などとも呼ばれています。

文字ファイルを作成するには、普通、エディタやワープロを使います。各種のソフトを開発する際、それぞれのプログラミング言語によるソースプログラム(プログラムを記述した文字ファイル)の作成にはエディタが使われます。また、ビジネス文書などの文字ファイルであればワープロが使われます。さらに、表計算ソフトやデータベースソフトなどのビジネスソフトからも、文字ファイル形式のデータが作成できる場合もあります。

TYPE コマンドを実行するための書式を次に示します。なお、「*x*：」はドライブ名で、それがカレントドライブの場合は省略できます。

```
A>TYPE  x：ファイル名↵
```

TYPE コマンドは内蔵コマンドであり、実行するためのプログラムは MS-DOS に内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。

なお、ワープロによって作成された文書ファイルは、ディスクに保存する際のファイル形式の選択によっては、純粋な文字だけのファイル(ここで対象にしている文字ファイル)ではなく、文字以外のいろいろな制御コードなど ―― たとえば、各種の文字飾り、罫線、文字間隔、行間隔などをプリンタに指示するためのコードや、各文書ファイルをワープロが管理するためのヘッダと呼ばれる部分など ―― が含まれるものがありますので注意が必要です。文字以外のコードを含んだ文書ファイルは、TYPE コマンドでは正常に表示することができません(エディタによって作成されるファイルはいずれも純粋な文字ファイルである)。

また、各種のビジネスソフトのデータファイルなども、ディスクに保存する際に、テキスト形式(文字ファイル形式)の指定をして保存しなければ、たいていは文字以外のコードを含んでいるでしょう。いずれにしても、純粋な文字ファイルでなければ、TYPE コマンドでの表示はできません。当然のことですが、「.COM」や「.EXE」のファイルタイプが付いたファイルは、コンピュータが実行するためのマシン語プログラムですので、TYPE コマンドで内容を読むことはできません。

さて、TYPE コマンドの使い方は、とくに説明する必要もないほど簡単ですので、次の実行例をご覧ください(図 7.6)。

```
A>TYPE README.DOC ↵
```
カレントドライブ(A:)上の文字ファイル「README.DOC」をディスプレイに表示する。TYPEコマンドのプログラムは内蔵コマンドなので、ディスク上になくてよい

1.　バックアップ作成時の注意

　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタフォーマット（７２０ＫＢ）でフォーマットされています。
　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、バックアップ媒体のフォーマット時に／９スイッチを指定してください。

（例）

このようにファイルの内容がディスプレイに出力される。CTRL＋P をONにしておけば、プリンタにも同時に出力させることができる。また１画面におさまらない場合は、CTRL＋S で表示を一時ストップすることができる

に添付されている「固定ディスクを使用する場合…
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけます。

```
A>B: ↵
```
カレントドライブをB:に移す

同様にTYPEコマンドを実行する

```
B>TYPE README.DOC ↵
ファイルが見つかりません.
```
TYPEコマンドは内蔵コマンドなので、カレントドライブに関係なく実行できる。ここでの「ファイル」とは、ディスプレイ表示の対象となる「README.DOC」のことをいっている。つまり、カレントドライブ(B:には「README.DOC」がない

```
B>TYPE A:README.DOC ↵
```
ファイルの頭にドライブ名を付けて「ドライブA:上のREADME.DOC」と指定した

1.　バックアップ作成時の注意

　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタフォーマット（７２０ＫＢ）でフォーマットされています。
　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、…時に／９スイッチを指定してください。

　固定ディスクの媒体は、極めて精密、均質に…記録されるため、読み出しエラーが起こりやすい場所（セクタ）ができることがあります。
　このようなセクタは、「スキップセクタ」と呼び、システムでデータ記憶領域として使用しないようにして、お客様のデータが失われることのないよう、予防しています。
　固定ディスクに対して、ＣＨＫＤＳＫコマンドを実行すると、「×××××バイト：スキップセクタ」と表示される場合がありますが、これは、このような予防処置がされていることを表しますので、「スキップセクタ」が表示されても安心してご使用いただけます。

　なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけます。

```
B>
```
ドライブA:上に存在する「README.DOC」が表示されている

— 図 7.6 —（次ページに続く）

```
A>TYPE D:MSDOS1.BUN
gf0#3#6#1#'(-,871002871013!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!y!C:¥3WPX121.B
UN    AVAeArAsAiAoAn!!A1A0A.A1A3=c!c!3#W#P#d!d!=3#%!*%U%#%9%G$N$o%<!W%m%xMQMcN=
4#%!o%<!W%m%N$H;$$}JA UI-$g9$$}J=5#%!o%<!W%m%&!=%U%H%N$*ArB=6#%!o>K$/H8EQ%=%3%
s%&!o%<!W%m%!!!...!!!C%9%/%H%C%W%&!Q%V%j%C%7%s%0%!!J$I$!!!!yDa$/$/$j$=!
.. A-A-A-A-A-A-A-=3#%!*%U%#%9%G$... ‥A-A-A-A-A-A-A-A-A-A-A-A-A-A-A-A-
W%m%r$+8F$$$k$H$"!A1|FfCY5'$V4b$J$/$"!~F1$...
H6r$7$F$$$^$9$#!o%<!W%m%N$p4¥K*EJ$!5=Gd$E;HA_$0$@$$$?$$$,Jj$^$7$?$,$"!d;?$A$0$=$
1$r$|Fo>N$S%8%M%9%K$I$N$h$&$K$h3+$7$F$$$1$P$h$$$N$G$7$g$&$#!!Q%=%3%s%&!o%<!W%m%0
$"!A10>N$|!G$R=Y$?$h$&$K$"![0J$k$!5o<"![0J$k$o%<!W%m%&!=%U%H%V4G$"$C$F$b$"!_8$$N
$8J0>U%!%$%k%r$U%m%C%T%<!G%#%9%/%G$>D¥@r8949$k$3$H$,$G$-$^$9$#!5$i$K$"!=IW7;;=%U
%H%d$G%<!?%Y%<!9%&!=%U%H%J$I$N$F3o<N$S%8%M%9%&!=%U%H%N$G%<!?%b$"!o%<!W%m%N$8J0>

A>
```

ワープロソフト（新松）により作成されたドライブD：上の文書ファイル「MSDOS1.BUN」をディスプレイに表示する

新松の場合、「.BUN」の付いた通常の文書ファイルは「テキスト形式」ではないので、TYPEコマンドでディスプレイに表示することはできない。ただし、一太郎の「.JXW」の付いた文書ファイルは「テキスト形式」であり、表示が可能である

```
A>TYPE FORMAT.EXE
MZ牝 (""f(n■3<岬)ｸﾑｽU駆軽ｷｽF■F■F~ｸ9P奪ﾐ
ﾀt鮴ﾐu3闘」ｸNP頚ｹﾐ隶冶"宇~闔冶2、×~#u髪ｸYP靠ｹﾐ髞孝■"6v"v鑚ﾗF~ﾀu[禦#?#u隰
  宇■噬F■■F=t.皆<u冶L宇~耶=+ﾀP"v■噬*膠隋ﾐ■驩F~v~ｸcP鍼ｹﾐ■yPｹﾐｽF"""祈~句
]ﾃU駆ｸH
軽ｶｽ■■F■F■D宇■t濶F■uｽ■■~u孝■"ｶB■祉ﾄ宇■t=u鶏=u鴬■号t■"v■ｵﾐ宇■t鯰号
■P号t■号ｸ■号T■号R■孝~P"v■■ﾐ宇■t骭「ｸ弁鑚ｸﾐ宄R■tH=u3号■P号T■号ｸ■孝~P
号t■号R■号B■号D■"v■Ⅲﾐ宇■"「ｸｧPｸﾐ"号■P号T■号ｸ■孝~P号t■号R■号B■号D■
"v■P<■ｸｱP韭ｷﾐｽF"鴆~ｸﾓ■ｸ■ｫｸ■ｸ)■CP猩ｷﾐｽF■祈■蘊ﾃU駆ｸ郁jｵVｽFｮｽF碑ｲA記杏F幕
v免BF麻~目|■FP皆*膠隶ﾐﾀt=u鴉=u羌=u鱧ｸ0鯇号|"P孝捻孝ｮP皆*膠勒βt魏ｽFｪ皆$<t
 皆$#<#u'+ﾀP"6L阡ﾐﾀuｽFｪ■Fｪ"""ｸPｶﾐ皆$<tFPｸ^P肇ｶﾐ鑰ｶｸ#P砲ｶﾐ皆*膠"vvｮ孝P号|"P阯
ﾐ
ﾀtﾐｪ"u鐶v韶ﾐ宇ｦ搭Fｦｸｾｸ蓁W闥    ﾀt鬯皆$<uKｸNPｸ」P鏤ｶﾐｸｼPｸｦP鏐ｶﾐ孝楙靠!ﾐ■、ｸｩ
P銭ｶﾐ孝椰陏ﾐﾀt
ｸｨPｶﾐ皆$<tG=F、=F-N#~、t饌"ｸP雑ﾐ孝-P闢ｳﾐ#~ｱuﾝ#~ｲYt5#~ｲyt/#~ｲnt#~ｲNu=F、"ｽFｪ"""ｸﾎ
P犇ｵﾐｽFｪ""■=F、■=F、"ｸ3P隰ｵﾐ孝Pｸ6P陞ｵﾐｵｸxP闥ｵﾐ雕"ｸ{P閦ｵﾐ■簿■ｯ■ｸﾝ■ｸ■^句
]ﾃU駆ｸv
""v閭ﾄ皆,A■~疑活F■F~8F■糾FF~r+ﾀ■句]ﾃU駆ｸZ勒ｲV皆 *-A宇ｨFｩ曲ﾑ狠愚v#<u#|uｽFｮ■
ｽFｮ■Fｮ疑蛙C
```

FORMATコマンド（マシン語でできている）を表示する

文字ファイル以外はTYPEコマンドの実行はできない

図 7.6 TYPEコマンドの実行例

7.5 ファイル名の変更 REN（リネーム）

― 内蔵コマンド、ワイルドカード使用可 ―

ファイルの名前は、REN コマンド(REName)を使って、自由に変更することができます。この REN コマンドは内蔵コマンドであり、実行するためのプログラムは MS-DOS に内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。

REN コマンドを実行するための書式を次に示します。「x：」はドライブ名で、カレントドライブの場合は省略できます。

```
A>REN  x：旧ファイル名  新ファイル名⏎
```

```
A>REN  x：旧ワイルドカード  新ワイルドカード⏎
```

これらは、ドライブx：上のファイル「旧ファイル名」を、「新ファイル名」に変更するものです。また、ファイル名にワイルドカード記号「?」「*」を使うこともできます。いずれも、旧ファイル名、新ファイル名の、左、右の順序を逆にしないように注意してください。左側から右側にファイル名が変わります。

REN 旧ファイル名 → 新ファイル名

では実行例を図 7.7に示します。実習用の適当な内容のディスクをセットして試してみてください。なお、ディスクが書き込み禁止(ライトプロテクトのシールが貼ってあるなど)になっている場合は、REN コマンドの実行はできませんので注意が必要です。

```
A>DIR/W⏎ ……………実習前のファイル名を確認する

 ドライブ A: のディスクのボリュームラベルはありません．
 ディレクトリは A:¥

COMMAND  COM    FORMAT   EXE    DISKCOPY EXE    CHKDSK   EXE    EDLIN    EXE
PRINT    SYS    MOUSE    SYS
        7 個のファイルがあります．
   980992 バイトが使用可能です．

A>REN CHKDSK.EXE CDSK.EXE⏎ ……………カレントドライブ(A:)上のファイル「CHKDSK.EXE」を
                                  「CDSK.EXE」に変更する

A>DIR/W⏎ ……………結果を確認する

 ドライブ A: のディスクのボリュームラベルはありません．
 ディレクトリは A:¥
```

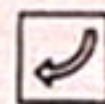
― 図 7.7 ―（次ページに続く）

```
COMMAND  COM    FORMAT   EXE    DISKCOPY EXE     CDSK     EXE     EDLIN    EXE
PRINT    SYS    MOUSE    SYS
        7 個のファイルがあります.
   980992 バイトが使用可能です.

A>CDSK
  1250304 バイト : 全ディスク容量
    95232 バイト : 2 個のシステムファイル
   174080 バイト : 7 個のユーザーファイル
   980992 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   426640 バイト : 使用可能メモリ

A>REN *.SYS *.DEV

A>DIR/W

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    FORMAT   EXE    DISKCOPY EXE     CDSK     EXE     EDLIN    EXE
PRINT    DEV    MOUSE    DEV
        7 個のファイルがあります.
   980992 バイトが使用可能です.

A>B:

B>REN FORMAT.EXE F.EXE
ファイル名が重複しているか, またはファイルが見つかりません.
B>REN A:FORMAT.EXE F.EXE

B>DIR A:/W

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    F        EXE    DISKCOPY EXE     CDSK     EXE     EDLIN    EXE
PRINT    DEV    MOUSE    DEV
        7 個のファイルがあります.
   980992 バイトが使用可能です.

B>A:F C:
Format Version 4.10

新しいディスクをドライブ C: に挿入し
どれかキーを押してください^C

B>
```


図 7.7 REN コマンドの実行例

7.6 ファイルのコピー(転送) COPY

－ 内蔵コマンド、ワイルドカード使用可 －

3.4 節において、ディスク上のファイルを COPY コマンドを使ってコピーしました。COPY コマンドは多くの場合、ディスク上のファイルのコピーに使われます。しかし COPY コマンドは、ディスク上のファイルをディスク上にコピーするだけでなく、各周辺装置間(ディスクもその中の1つであるが)でデータをやり取りする機能を持つ「データ転送コマンド」として捉えなければなりません。つまり COPY コマンドは、

- ディスク
- キーボード
- ディスプレイ
- プリンタ
- RS-232C インターフェイス
- その他の周辺装置

などの周辺装置間を相互に結ぶデータ転送のためのプログラムなのです。たとえば、

ディスク上のファイル	↔	ディスク上のファイル
ディスク上のファイル	↔	RS-232C インターフェイス
ディスク上のファイル	→	プリンタ
キーボード入力	→	ディスク上のファイル
キーボード入力	→	RS-232C インターフェイス
キーボード入力	→	プリンタ

などのデータ転送を行うことができます(図 7.8)(周辺装置間のデータ転送については、9 章で詳しく解説する)。また、COPY コマンドには付加機能として、ディスク上のファイルをコピーする際、いくつかのファイルをリンク(連結)して、連続した1本のファイルにするなどの機能もあります。

COPY コマンドを実行するための書式を次に示します。「x：」「y：」はドライブ名で、カレントドライブの場合は省略できます。ただし、ここでの「ファイル名」とは、ディスク上のファイルの名前だけではなく、「コンソール」(通常は、キーボードとディスプレイのことを指す)や「プリンタ」などの周辺装置を含めた広義の意味です(ただしディスク装置以外の場合はドライブ名は付かない)。

```
A＞COPY  x：送出側ファイル名  y：↵
```

```
A＞COPY  x：送出側ワイルドカード  y：↵
```

```
A＞COPY  x：送出側ファイル名１  y：受取側ファイル名２↵
```

```
A＞COPY  x：送出側ワイルドカード１  y：受取側ワイルドカード２↵
```

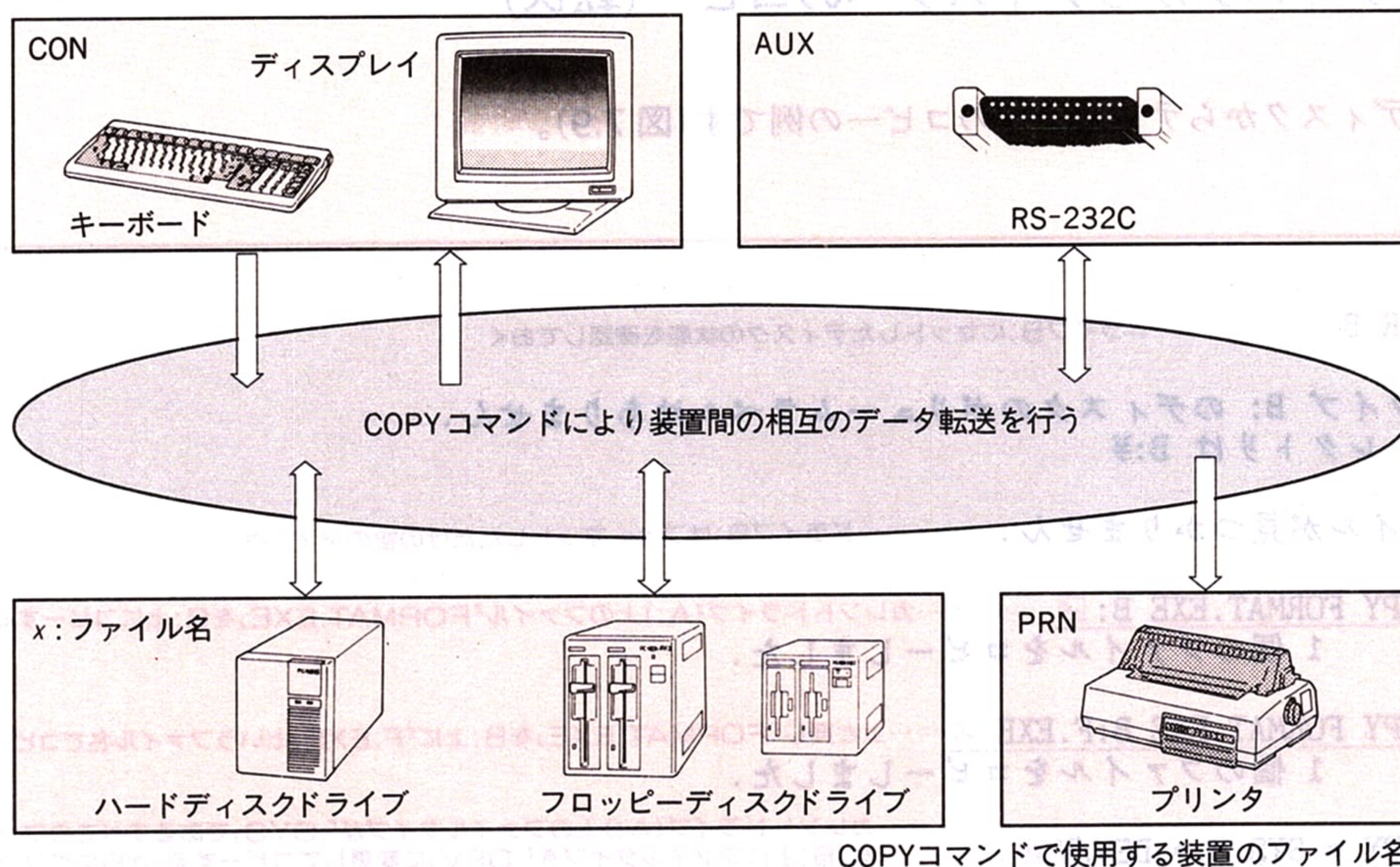



図7.8　COPYコマンドによる各周辺装置間のデータ転送機能

さらにスイッチによる次の付加機能がありますが、その他の付加機能については、本書では触れません。なお、いずれもこれらの書式の最後に付け加えます。

/V …… ベリファイスイッチ。ディスク間のファイルのコピーの際、書き込んだデータをその場で読み出して、元データと比較し、正しい書き込みが行われたかをチェックする機能

では、COPYコマンドの多くの使い方の中から、

ディスク上のファイル　→　ディスク上のファイル
キーボードからの入力　→　ディスク上のファイル
ディスク上のファイル　→　プリンタ

への転送(コピー)の実行例を示しましょう。COPYコマンドは内蔵コマンドであり、実行するためのプログラムはMS-DOSに内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。

7.6.1　ディスクからディスクへのコピー(転送)

まず、ディスクからディスクへのコピーの例です(図7.9)。

```
A>DIR B:↵ ……………ドライブB:にセットしたディスクの状態を確認しておく

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

ファイルが見つかりません. ……………ドライブB:はフォーマットしただけの空のディスク

A>COPY FORMAT.EXE B:↵ ……………カレントドライブ(A:)上のファイル「FORMAT.EXE」をB:上にコピーする
        1 個のファイルをコピーしました.

A>COPY FORMAT.EXE B:F.EXE↵ ……上と同じ「FORMAT.EXE」をB:上に「F.EXE」というファイル名でコピーする
        1 個のファイルをコピーしました.

A>COPY *.SYS B:*.DEV/V↵ ……カレントドライブ(A:)上のファイルタイプが「.SYS」であるすべてのファイル
                           を、B:上にファイルタイプを「.DEV」に変更してコピーする。さらにベリファイ
                           スイッチ「/V」を付けて、コピー直後にそれを読み出して比較チェックをさせ、コ
                           ピーされたデータが正しいことを確認させる
PRINT.SYS  } コピーされるファイル名
MOUSE.SYS  } が順次表示される
        2 個のファイルをコピーしました.

A>DIR B:↵ ……………ドライブB:のファイルを確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

FORMAT   EXE    97766  88-07-13    0:00 } この2つは、ファイル名が違っても内容はまったく同じもの
F        EXE    97766  88-07-13    0:00 }
PRINT    DEV     5855  88-07-13    0:00 } ワイルドカード(*)を使ってコピーされたもの。
MOUSE    DEV     3985  88-07-13    0:00 } ファイルタイプが「.DEV」に変更されている
        4 個のファイルがあります.          データの信頼性は/Vスイッチにより保証されている
  1043456 バイトが使用可能です.

A>COPY B:F.EXE A:FMT.EXE↵ ……………ドライブB:上のファイル「F.EXE」をA:上に「FMT.EXE」
                                    というファイル名でコピーする
        1 個のファイルをコピーしました.

A>DIR/W↵ ……………ドライブA:のファイルを確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥
```

— 図7.9 — (次ページに続く)

```
                  ワイルドカード(＊)の対象となった「.SYS」ファイル
COMMAND  COM      FORMAT   EXE      DISKCOPY EXE      CHKDSK   EXE      EDLIN    EXE
PRINT    SYS      MOUSE    SYS      FMT      EXE
        8 個のファイルがあります.
   882688 バイトが使用可能です.
                                    ドライブB:上の「F.EXE」がコピー
A>                                  されたもの。そのもとをたどれば左上
                                    の「FORMAT.EXE」である
```

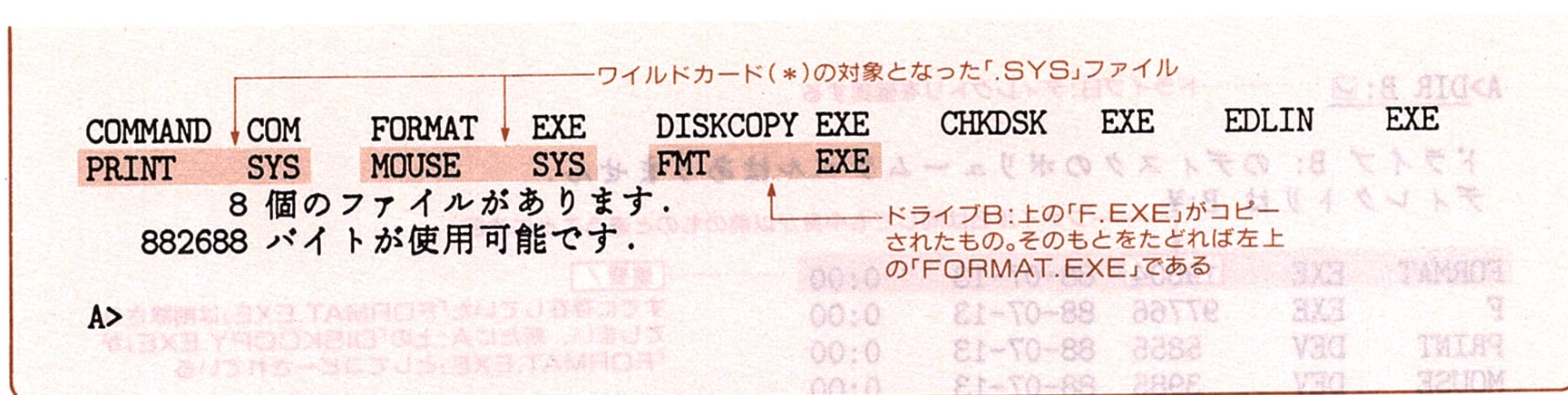



図 7.9　ディスクファイルからディスクファイルへのコピー例

COPY コマンドは、コピー元のディスクを交換したり、受け取る側のディスクを交換したりしながら作業を続けることができます。つまり、複数のコピー元のディスクから任意のファイルを選択してコピーし、必要なファイルだけを任意のディスクに集めたりすることができるわけです。

ただし、ディスク上のファイルのコピーには十分注意しなければなりません。次の実行例に示すように、あるファイル(たとえば ABCD.XYZ)を別のディスクにコピーする場合、コピー先のディスク上に、同じファイル名のファイル「ABCD.XYZ」がすでに存在していると、もとあったファイルは削除され、同名の新たにコピーされたファイル(新しい ABCD.XYZ)に置き換わってしまいます。

このようなことは、ファイル「ABCD.XYZ」を別のファイル名(たとえば 1234.XYZ)としてコピーする場合でも同じです。そのファイル名(1234.XYZ)が、すでにコピー先のディスク上に存在している場合は、もとあったファイル「1234.XYZ」は削除され、同名の新たにコピーされたファイル(内容は ABCD.XYZ)に置き換わってしまいます。その実行例を図 7.10 に示します。

```
A>DIR B:↵ ……………ドライブB:のファイルを確認しておく

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥
                  このファイルサイズに注目しておく
FORMAT   EXE    97766  88-07-13    0:00  この2つは、ファイル名が異なるだけで、同じもの。
F        EXE    97766  88-07-13    0:00  次の実行結果の確認のために注目しておく
PRINT    DEV     5855  88-07-13    0:00
MOUSE    DEV     3985  88-07-13    0:00
        4 個のファイルがあります.
  1043456 バイトが使用可能です.

A>COPY DISKCOPY.EXE B:FORMAT.EXE↵ ……………ドライブB:上にすでに「FORMAT.EXE」が存在して
        1 個のファイルをコピーしました.      いる場合に、A:上の「DISKCOPY.EXE」をファイル
                                             名「FORMAT.EXE」としてB:上にコピーする
```

— 図 7.10 —（次ページに続く）

```
A>DIR B:⏎ ……………ドライブB:ディレクトリを確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥
                ↑—ファイル名は同じでも中身が以前のものと違うことに注目

FORMAT   EXE    19604  88-07-13   0:00  ……………重要!
F        EXE    97766  88-07-13   0:00  すでに存在していた「FORMAT.EXE」は削除され
PRINT    DEV     5855  88-07-13   0:00  てしまい、新たにA:上の「DISKCOPY.EXE」が
MOUSE    DEV     3985  88-07-13   0:00  「FORMAT.EXE」としてコピーされている
        4 個のファイルがあります.
  1121280 バイトが使用可能です.

A>B:FORMAT B:⏎ ……………「FORMAT」を実行してみる

DISKCOPY version 4.00

ディスクのコピーを行います

送り側ディスクをドライブ B: に挿入してください  このように、「FORMAT」コマンド
受け側ディスクをドライブ A: に挿入してください  ではなく、「DISKCOPY」コマン
準備ができたらどれかのキーを押してください^C   ドが実行される
                                            ↑
A>                                          └—CTRL+C を入力してコマンドレベルに戻る
```

図7.10　コピーにより作成しようとするファイル名がすでに存在している場合のCOPYコマンドの実行例

■ディスク上のファイルの連続性について

さてここで、ディスク上のファイルを別のディスクにコピーする場合、もとのディスクと、コピーにより作成されたディスクとの、**ファイルの連続性**の違いについて解説しましょう。

ディスク上にいくつものファイルが存在し、それらのファイルの削除や変更、それに新たなファイルの作成やコピーなどの各種の操作が何回も行われたとしましょう。そのようなディスク上に書き込まれているたいていのファイルの状態は、それぞれのファイルの物理的な連続性が断たれています。

たとえば、そのようなファイルの状態のディスクに、1つの新しいファイルを書き込む場合、MS-DOSはそのディスク上のあちこちに散在している空き部分を捜しながら、それらを埋めるように書き込んでいきます。したがって、コピーしたファイルは全体が連続せず、いくつかのブロックに分かれてディスク上のあちこちに格納されることになります(**図7.11**)。

←ディスク外周部　ディスク内周部→

状態(1)　ファイル1　ファイル2　ファイル3　ファイル4　ファイル5　空のディスクに5つのファイルをコピーした。それぞれ連続したファイルがつながって格納されている

(2)　空　ファイル2　ファイル3　空　ファイル5　ファイル1、ファイル4を削除した

(3)　ファイル6(a)　ファイル2　ファイル3　ファイル6(b)　空　ファイル5　そこにファイル6をコピーした。(a)と(b)の2つに分割されて格納された

(4)　ファイル6(a)　ファイル2　空　ファイル6(b)　空　ファイル3、ファイル5を削除した

(5)　ファイル6(a)　ファイル2　ファイル7　ファイル8(a)　ファイル6(b)　ファイル8(b)　そこにファイル7、ファイル8をコピーした。ファイル8は(a)と(b)の2つに分割されて格納された

COPYコマンドで別の空ディスクにコピー

(6)　ファイル6　ファイル2　ファイル7　ファイル8　COPYコマンドで別の空ディスクにコピーした状態。分割されていたファイルは連続したファイルとしてコピーされている

※1枚のディスクの全トラックを1本につなげて模式的に表したもの

図7.11　ディスク上のファイルの散在と連続

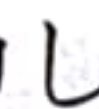
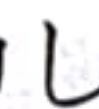
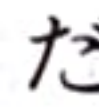
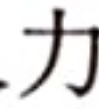
そのファイルの散在と連続の状態を示したのが図7.11ですが、この図では、ディスクの全トラックを、連続した1本の横の棒として表現しています。左端がディスク外周部、右側が内周部に向かっています(実際のトラックは、数十本～百数十本の同心円状であることは、3.1節で解説した)。実際に、ファイルが散在しているか、連続しているかは、CHKDSKコマンドを使って確認することができますが、その実習は7.9節で行います。

ただし、ファイルの物理的な記録状態がこのように連続していなくても、異常ではありません。いろいろな作業を行っているディスクでは、むしろ連続していない方が普通です。その場合、散在しているファイルの読み出し／書き込みは、連続しているファイルに比べて多少遅くなりますが、その他には何の影響もなく、気にする必要はありません。

ところが、連続していない複数ファイルを、COPYコマンドを使って別の空のディスク(空であるこ

とが重要)にコピーすると、それぞれのファイルは、書き込まれるディスクの外周部から順に、連続した状態で格納されます。このことから、いろいろな作業をさんざん行ったディスクの場合、適当なときにCOPYコマンドを使って、たとえば、

```
A>COPY A:*.* B:⏎
```

のように、別の空ディスクにコピーし直すと、それぞれのファイルが、きれいに連続して書き込まれ、その後の読み出し／書き込みが速くなります。もちろん「*.*」(すべてのファイル)とせずに、必要なファイルだけをコピーしてもかまいません。

当然ながら、そのためにDISKCOPYコマンドを使ったのでは意味がありません。DISKCOPYコマンドは、ディスクのまるごとコピーなので、ファイルが散在していれば、そのままの状態でコピーされます。

7.6.2　キーボードからディスクへのコピー(転送)

さて、COPYコマンドに関するこれまでの実習は、ディスク上のファイルをディスク上にコピーするものでした。しかし本章の冒頭で述べたように、COPYコマンドは本来、周辺装置間のデータ転送コマンドです。その実際を、次の簡単な実行例で示してみましょう(周辺装置間のデータ転送については、再度9章で解説する)。

最初は、**CON**(コンソール)から、ディスクファイルへの転送例です。つまり、キーボードから入力した文字を、ファイルとしてディスクに格納します。これは、COPYコマンドを使って、文字ファイルを作成できることを意味します。

文字ファイルを作成するには、普通、エディタやワープロなどを使いますが、簡単で短い内容(たとえば、自動スタート用のバッチファイル「AUTOEXEC.BAT」(10.2節で解説)や、コンフィギュレーションファイル「CONFIG.SYS」(APPENDIXで解説))であれば、このCOPYコマンドを利用して手軽に作成することができます。

ところで、COPYコマンドを使って文字ファイルを作成する場合、入力する文字列の各行の終わりには⏎をキー入力しますが、この⏎のキー入力後は、それまで入力した文字の訂正はできませんので慎重に作業してください。各行各行の⏎をキー入力する前には、誤りがないかを十分チェックする必要があります。不運にも、⏎してから誤りを発見した場合は、最初のCOPYコマンドの入力からやり直しです。具体的には、CTRL＋Cをキー入力してCOPYコマンドの実行を中止し、MS-DOSのコマンドレベルに戻ってから、再度挑戦することになります。(図7.12)

すべての文字(文)の入力が終われば、CTRL＋Z(CTRLキーを押しながらZを入力する)を入力したあと⏎を入力することにより、COPYコマンドが終了し、最初に指定した名前のファイルが作られます。

```
A>COPY CON B:TESTFILE.ABC ↵ ········コンソールCON(キーボード)から入力したものを、ドライブB:上にファイル
                                    「TESTFILE.ABC」としてコピーする。つまり新しいファイルを作成する
Where have all the flowers gone ↵
Long time passing ↵                  適当な文を入力していく。カナ文字や日本語入力機能
Where have all the flowers gone ↵    による漢字などを入力してもよい
Long time ago ↵                      (これはだいぶ昔のフォークソング)
^Z ↵ ··············最後は CTRL + Z を入力する。「^」+「Z」ではないので注意
        1 個のファイルをコピーしました.

A>DIR B:TESTFILE.ABC ↵ ··············ドライブB:上に作成されたファイル「TESTFILE.ABC」を確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

TESTFILE ABC      102  89-07-25   15:30 ··············102バイト(復帰・改行も含めて102半角文字)の
        1 個のファイルがあります.                         文字ファイルが作成されている
  1249280 バイトが使用可能です.

A>TYPE B:TESTFILE.ABC ↵ ··············その内容をディスプレイに表示する

Where have all the flowers gone
Long time passing                    コンソール入力(キー入力)がこのように文字ファイルとなっている
Where have all the flowers gone
Long time ago

A>
```

図 7.12　キー入力データをディスクファイルに格納する例

7.6.3　ディスクからプリンタへのコピー(転送)

次に、ディスク上のファイルを PRN(プリンタ)に転送する例を示しましょう。つまり、ディスクファイルの内容をプリンタへ出力します(図 7.13)。ただし、ファイルは文字ファイルでなければなりません。ここでは、図 7.12 で作成したファイルを対象にしてみましょう*。

---

* PC-9800 シリーズの MS-DOS バージョン 3.*x* 以降の場合、MS-DOS のコマンドレベルにおいてプリンタを動作させるには、プリンタのデバイスドライバ(装置を動作させるためのプログラム)を組み込んでおく必要がある。5.3 節を参照のこと。

```
A>COPY B:TESTFILE.ABC PRN⏎
        1 個のファイルをコピーしました.

A>
```

ドライブB:上の文字ファイル「TESTFILE.ABC」を、周辺装置PRN(プリンタ)にコピーする(つまりプリンタに出力する)

⬇

プリンタにはこのように出力される

```
Where have all the flowers gone
Long time passing
Where have all the flowers gone
Long time ago
```

図 7.13　ディスクファイルをプリンタに出力する例

7.7　ファイルの削除　DEL(デリート)

― 内蔵コマンド、ワイルドカード使用可 ―

ディスク上の不要なファイルを削除するには DEL コマンド(DELete)を使います。ワイルドカードが使えますので、特定のグループのファイルを一度に削除することが可能です。

ファイルの削除といっても、内部的には、該当するファイルのディスク上の全内容を、すべて物理的に消去してしまうわけではありません。実際に行われる動作は、ディスクのディレクトリ領域(4.1、4.2 節参照)にある、ファイルの「登録台帳」から、該当するファイルの登録が抹消されるだけです。つまり、本人(内容)はそのままで戸籍を抹消されてしまうわけです。

ただしこれは内部的な話であり、私たちユーザーにとっては、DEL コマンドを実行すれば、対象となったファイルのすべてがディスク上から消えてなくなると考えてけっこうです。もちろん削除されたファイルが占めていたディスク上の領域は解放され、使用可能容量がその分増えることになります。

DEL コマンドを実行するための書式を次に示します。「*x*：」はドライブ名で、カレントドライブの場合は省略できます。

```
A>DEL  x：ファイル名⏎
```

```
A>DEL  x：ワイルドカード⏎
```

では実行例を示しましょう(図 7.14)。このDELコマンドは内蔵コマンドであり、実行するためのプログラムはMS-DOSに内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。なお、このコマンドの実習には、どのファイルを削除しても差し支えのないような実習用のディスクを使ってください。

```
A>DIR/W↵ ……………ドライブA:上のディスクの状態を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM     CONFIG   SYS     ATOK6A   SYS     ATOK6B   SYS     ATOK     DIC
PRINT    SYS     RSDRV    SYS     MOUSE    SYS     FORMAT   EXE     DISKCOPY EXE
CHKDSK   EXE     SYS      EXE     FC       EXE     FIND     EXE     MORE     COM
SORT     EXE     TREE     EXE     ATTRIB   EXE     EDLIN    EXE     README   DOC
AUTOEXEC BAT
        21 個のファイルがあります.                    MS-DOSの常用システムディスクである
    357376 バイトが使用可能です.

A>DIR B:/W↵ ……………ドライブB:のディスクの状態を確認する

 ドライブ B: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは B:¥

COMMAND  COM     FORMAT   EXE     MP       COM     MP       COD     MP       SYS
MP       DAT     MP       HLP     MP       PIF     MOUSE    SYS     SETMP    INI
601_P    CPD     601_DL   CPD     601_DP   CPD     601_L    CPD     201H     CPD
201H_D   CPD     201H_DS  CPD     201H_S   CPD     ATOK6A   SYS     ATOK6B   SYS
ATOK     DIC     CONFIG   SYS     AUTOEXEC BAT     MP       INI
        24 個のファイルがあります.
     34816 バイトが使用可能です.                      Multiplanの常用ディスクである

A>
```

```
A>DEL *.EXE↵ ……………カレントドライブ(A:上)の、ファイルタイプが「.EXE」であるファイルをすべて削除する

A>DIR/W↵ ……………結果を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM     CONFIG   SYS     ATOK6A   SYS     ATOK6B   SYS     ATOK     DIC
PRINT    SYS     RSDRV    SYS     MOUSE    SYS     MORE     COM     README   DOC
AUTOEXEC BAT
        11 個のファイルがあります. ……………11個になった。「.EXE」ファイルは削除されている
    566272 バイトが使用可能です.

A>DEL *.COM↵ ……………同様にすべての「.COM」ファイルを削除する
```

— 図 7.14 — (次ページに続く)

```
A>DEL README.DOC ↵ ……………「README.DOC」を削除する

A>DIR/W ↵ ……………結果を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

CONFIG   SYS    ATOK6A   SYS    ATOK6B   SYS    ATOK     DIC    PRINT    SYS
RSDRV    SYS    MOUSE    SYS    AUTOEXEC BAT
        8 個のファイルがあります. ……………8個になった。「.COM」ファイルと「README.DOC」は
   604160 バイトが使用可能です.                    削除されている

A>DEL B:MP.* ↵ ……………ドライブB:上の主ファイル名が「MP」であるすべてのファイルを削除する

A>DIR B:/W ↵ ……………結果を確認する

 ドライブ B: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは B:¥

COMMAND  COM    FORMAT   EXE    MOUSE    SYS    SETMP    INI    601_P    CPD
601_DL   CPD    601_DP   CPD    601_L    CPD    201H     CPD    201H_D   CPD
201H_DS  CPD    201H_S   CPD    ATOK6A   SYS    ATOK6B   SYS    ATOK     DIC
CONFIG   SYS    AUTOEXEC BAT
       17 個のファイルがあります. ……………「MP. xxx」ファイルは削除されている
   411648 バイトが使用可能です.

A>DEL B:*.* ↵ ……………ドライブB:上のすべてのファイルを削除する
よろしいですか <Y/N>? Y ↵ ……………事が重大なので確認の問い合わせがある。「Y」を入力する

A>DIR B: ↵ ……………結果を確認する

 ドライブ B: のディスクのボリュームラベルは MULTIPLAN
 ディレクトリは B:¥

ファイルが見つかりません. ……………ドライブB:上のディスクは、すべてのファイルが削除された

A>DEL ABCD.XYZ ↵ ……………カレントドライブ(A:)上のファイル「ABCD.XYZ」を削除する
ファイルが見つかりません. ……………そのようなファイルは存在しない

A>DEL *.* ↵ ……………カレントドライブ(A:)上のすべてのファイルを削除する
よろしいですか <Y/N>? Y ↵ ……………事が重大なので問い合わせがある。「Y」を入力する

A>DIR ↵ ……………結果を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

ファイルが見つかりません. ……………ドライブA:もすべてのファイルが削除された

A>
```

図 7.14　DEL コマンドの実行例

7.8 ファイルを「削除／更新 禁止ファイル」にする「通常ファイル」に戻す ATTRIB

アトリビュート

－ 外部コマンド、ワイルドカード使用可 －

ATTRIB コマンドは MS-DOS のバージョン 3.*x* から提供されているコマンドです。

私たちは日常、さまざまなファイルを取り扱っていますが、その中には、絶対に削除されたり書き換えられては困る、重要なファイルもあるでしょう。それらのファイルを、削除や更新ができない「性質・属性」(アトリビュート)のファイルにしてしまい、MS-DOS のコマンドや、各種のユーザープログラムによる削除や更新から保護するためのコマンドが ATTRIB です。ただし、REN コマンドなどでリネームすることは可能です。したがって、エディタやワープロなどで更新する場合、それらがもとのファイルを「*.BAK」などにリネームしてバックアップファイルとして残すように設定されていると、最初の更新だけは行われてしまいますので注意してください(その「.BAK」ファイルは削除できないので、結局、ファイルの中身は保護されるが)。

MS-DOS 上のファイルには、「リードオンリーファイル」(読み出しのみ可能ファイル)とか、「ヒドゥンファイル(隠しファイル)」「システムファイル」などの、いくつかの「アトリビュート」(性質・属性)を持たせることができます。ATTRIB コマンドは、その中の「リードオンリーファイル」(つまり削除／変更 不能ファイル)のアトリビュートを付加したり、それを解除したりすることができ、また、現在付加されているアトリビュートの種類を表示することもできます。

ATTRIB コマンドの書式を次に示します。「*x*：」はドライブ名で、カレントドライブの場合は省略できます。

```
A>ATTRIB  ±R  x：ファイル名⏎
```

```
A>ATTRIB  x：ファイル名⏎
```

```
A>ATTRIB  ±R x：ワイルドカード⏎
```

```
A>ATTRIB  x：ワイルドカード⏎
```

1つ目の書式の「+R」は、ドライブ*x*：上のファイル「ファイル名」をリードオンリーファイルにするもので、「−R」は、リードオンリーファイルを解除するものです。2つ目の書式(「±R」を付けない場合)は、ドライブ*x*：上のファイル「ファイル名」のアトリビュートを調べる(表示する)ものです。この書式のコマンドを実行した結果、目的のファイル名の先頭に「R」が表示されるファイルは、リードオンリーのアトリビュートが設定されているファイル(つまり削除／変更 禁止ファイル)です。なお、リードオンリーのアトリビュートが付いたファイルを、COPY コマンドなどでコピーすると、コピー先のアトリビュートは解除されますので注意してください。ATTRIB コマンドのその他の機能

については省略します。

ATTRIB コマンドは、内蔵コマンドではなく外部コマンドです。これを実行するには、プログラムファイル「ATTRIB.EXE」がディスク上に必要です。では実行例を示しましょう(図 7.15)。

```
A>ATTRIB +R FORMAT.EXE ……………フォーマットコマンドの「FORMAT.EXE」ファイルに、
                                 リードオンリーのアトリビュートを付ける
A>DEL FORMAT.EXE ……………「FORMAT.EXE」ファイルの削除を試みる
アクセスは拒否されました. ……………削除できない

A>DIR FORMAT.EXE ……………「FORMAT.EXE」の存在を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

FORMAT   EXE    97766  88-07-13    0:00 ……………確かに存在している
        1 個のファイルがあります.
   357376 バイトが使用可能です.

A>ATTRIB FORMAT.EXE ……………「FORMAT.EXE」ファイルのアトリビュートを調べる
R A     A:¥FORMAT.EXE
←―――――――――――――――この「R」がリードオンリーファイルであることを示す
A>ATTRIB -R FORMAT.EXE ……………「FORMAT.EXE」ファイルのリードオンリーのアトリビュートを解除する

A>ATTRIB FORMAT.EXE ……………「R」が表示されない。リードオンリーのアトリビュートが
  A     A:¥FORMAT.EXE            解除され、削除や書き換えが可能になった

A>
```

図 7.15　ATTRIB コマンドの実行例

7.9　ディスクの内容／状態のチェック　CHKDSK(チェックディスク)

― 外部コマンド、ワイルドカード使用可 ―

CHKDSK コマンドは、本章の 7.3 や、4.1 節、4.2 節などでも使っていますので、その機能の一部はすでにご承知のことと思います。CHKDSK コマンドは、ディスク上のファイルの登録状況(11 章で解説する「階層ディレクトリ」の状況を含めて)や、ディスクの残り容量などを表示するだけでなく、ファイルの登録台帳や住所録などに当るディレクトリを検査し、もし何らかのエラーがあればその状況を表示したり、そのエラーが修復可能なものであれば、自動的に修復する機能があります。本書では、このエラーの修復のことについては触れませんが、通常の取り扱いをしていれば、ディレクトリのエ

ラーが発生するような事態はまずないでしょう。なお、ディレクトリのエラーが発生する可能性は、次のような場合に考えられます。

- ディスクの書き込みが行われている最中に電源をOFFにしたり、リセットボタンを押したり、ディスクのフタを開けてディスクを取り出してしまったような場合
- 書き込みを伴うプログラムを終了する際、決められたとおりの正常な終わり方をしなかった場合
- バグのあるプログラムを実行した場合

原因の多くはこのような場合 —— 専門的にいえばオープンしたファイルを正常にクローズできなかった場合でしょう。ただしエラーが発生したファイルがあっても、たいていの場合、ディスク全体や、ほかの正常なファイルの使用には影響はないでしょう(使用可能ディスク領域が減少する場合もあるが)。

CHKDSKコマンドの書式を次に示します。「*x*：」はドライブ名で、カレントドライブの場合は省略できます。

```
A>CHKDSK  x：↵
A>CHKDSK  x：ファイル名↵
A>CHKDSK  x：ワイルドカード↵
```

さらに次のスイッチによる付加機能があります。使い方は、上記の書式の最後部に付け加えます。

/V …… 階層ディレクトリの構造と、すべての階層に含まれるファイルの状況などを表示する
/F …… ディレクトリのエラーがあれば、修復可能なものは、自動的に修復する

階層ディレクトリ構造については、11章で説明します。11章は、今後MS-DOSを活用していくために、ほかの有力OSにも共通した、非常に重要な部分ですので、のちほどしっかり目を通してください。/Vスイッチについては、そこで実行例を示します。

また、2つ目と3つ目の形式は、指定したファイルが、ディスク上で物理的に連続したファイルになっているかどうかがチェックされます。ファイルの連続性に関しては7.6節で解説しましたが、これは連続していても、していなくても、どちらも正常なファイルです。

では実行例を示しましょう(図7.16)。CHKDSKコマンドは、内蔵コマンドではなく、外部コマンドですので、これを実行するには、ディスク上にプログラムファイル「CHKDSK.EXE」(あるいは.COM)が必要です。

```
A>CHKDSK B:⏎
```
……ドライブB:上のディスクの状態をチェックする。この場合、ドライブA:、B:は1Mタイプのディスクドライブが接続されている

```
  1250304 バイト : 全ディスク容量
   424960 バイト : 25 個のユーザーファイル
   825344 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   426624 バイト : 使用可能メモリ
```
825344 バイト……残っている使用可能な容量

655360／426624 バイト……コンピュータとメインメモリの容量

```
A>CHKDSK B:CHAP1⏎
```
……ドライブB：上のファイル「CHAP1」の状態をチェックする

```
  1250304 バイト : 全ディスク容量
   424960 バイト : 25 個のユーザーファイル
   825344 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   426624 バイト : 使用可能メモリ
```
上と同じものが再度表示される

```
B:¥CHAP1
     は 2 個の不連続ブロックを含みます.
```
ファイル「CHAP1」はディスク上で物理的に連続していない2個のブロックとして、ディスク上に散在している。ただしこれは「エラー」ではない。このファイルのリード／ライトが多少遅くなるだけの話である

```
A>CHKDSK B:*.*⏎
```
……ドライブB:上のすべてのファイルの状態をチェックする

```
  1250304 バイト : 全ディスク容量
   424960 バイト : 25 個のユーザーファイル
   825344 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   426624 バイト : 使用可能メモリ

B:¥CHAP1.BAK
     は 2 個の不連続ブロックを含みます.
B:¥CHAP9
     は 3 個の不連続ブロックを含みます.
B:¥CHAP5
     は 2 個の不連続ブロックを含みます.
B:¥CHAP12
     は 2 個の不連続ブロックを含みます.
B:¥CHAP6
     は 2 個の不連続ブロックを含みます.
B:¥CHAP5.BAK
     は 2 個の不連続ブロックを含みます.
B:¥CHAP4
     は 2 個の不連続ブロックを含みます.
B:¥CHAP1
     は 2 個の不連続ブロックを含みます.

A>
```
このディスクはワープロの文書用ディスクであり、作成、削除、変更を繰り返しているので、ほとんどのファイルが連続していない。
このようなディスクは、適当な時期に、空のディスクにCOPYコマンドでコピーすれば、連続したファイルとしてコピーされる。そうすればディスクの読み出し／書き込みが多少速くなるが、この「不連続」に関しては神経質になる必要はない

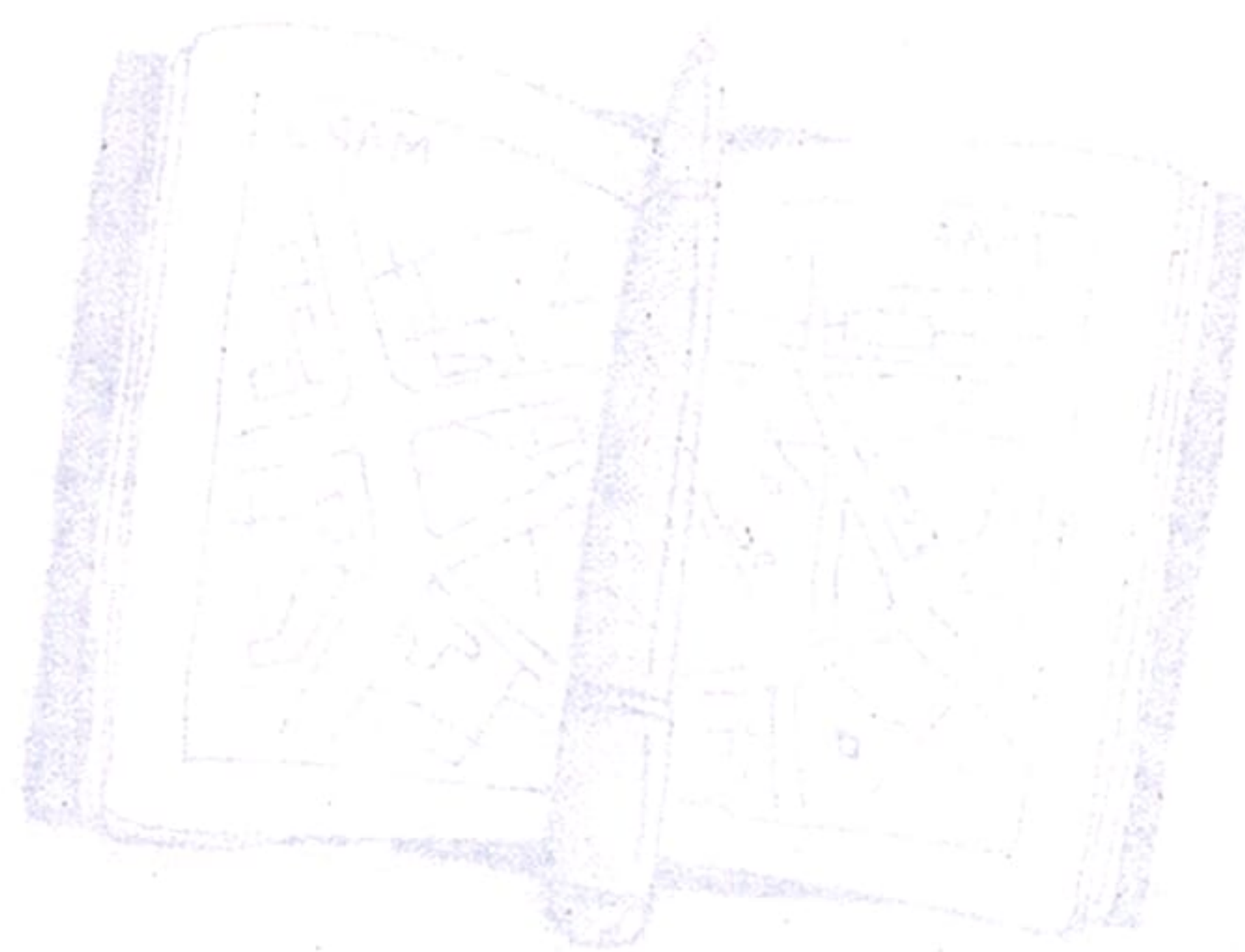
図7.16　CHKDSKコマンドの実行例

7.10 日付、時刻の表示および変更 DATE（デート）、TIME（タイム）

― 内蔵コマンド ―

この2つのコマンドは、MS-DOS を起動したときに、日／時の入力を要求してくるのと同じものです。つまり、MS-DOS が起動した時には、この DATE コマンドと TIME コマンドが内部で自動的に実行されるわけです。このコマンドについては、とくに解説の必要もないと思いますので、さっそく実行例を示します(図 7.17)。これらは内蔵コマンドであり、実行するためのプログラムは、MS-DOS に内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。

```
A>DATE↵ ……………DATEコマンドを実行する
現在の日付は 1989-07-27 (木) です.
日付を入力してください: 2001-4-1↵ ……………変更するには上の行に表示されているものと
                                              同じ形式で「年-月-日」を入力すればよい
                     ┌──変更されている
A>DATE↵              ↓
現在の日付は 2001-04-01 (日) です.
日付を入力してください:↵ ……………↵のみを入力すれば、変更はされずにコマンドレベルに戻る

A>TIME↵ ……………TIMEコマンドを実行する
現在の時刻は 14:30:28.00 です.
時刻を入力してください: 12:00:00↵ ……………変更するには上の行に表示されているものと
                                              同じ形式で「時:分:秒」を入力すればよい
                     ┌──変更されている
A>TIME↵              ↓
現在の時刻は 12:00:01.00 です.
時刻を入力してください:↵ ……………↵のみを入力すれば、変更はされずにコマンドレベルに戻る

A>
```

図 7.17 DATE、TIME コマンドの実行例

7.11 2つのディスク内容の比較 /Vスイッチ付 DISKCOPY（ディスクコピー）

― 外部コマンド ―

2つのディスクの内容がそっくり同じかどうか、DISKCOPYコマンドの/V(ベリファイ)スイッチにより比較することができます。ただし、機種によっては、DISKCOPYコマンドに、この機能がない場合もありますので注意してください。

このコマンドは、ディスク全面の物理的な記録内容が同じかどうかを、2つのディスクのトラック対トラックで比較していきます。つまり、3.2節で実習したDISKCOPYコマンドによる「まるごとコピー」により作成されたディスクであれば、この比較は完全に一致するわけです。次節で解説するFCコマンド(ファイル対ファイルの比較コマンド)による、2つのファイルの比較とは根本的に異なりますので混同しないようにしてください。

比較する2つのディスクは、記録容量、トラック数、セクタ数などが同じものでなくてはなりません。これは3.2節で解説した、DISKCOPYコマンドが使えるディスクの関係と同じですので、そこで示した表3.1を参照してください。

では実行例を示しますが(**図7.18**)、くれぐれも/Vスイッチを付けるのを忘れないように注意してください。忘れたことに気付いた場合は、最終的な実行の確認の問い合わせの際に CTRL ＋ C で実行を中断させることはできますが、それも忘れてしまった場合は、通常のDISKCOPYコマンドが実行され、実際にコピーが行われてしまいます。取り返しのつかないことにもなりかねませんので慎重に行ってください。

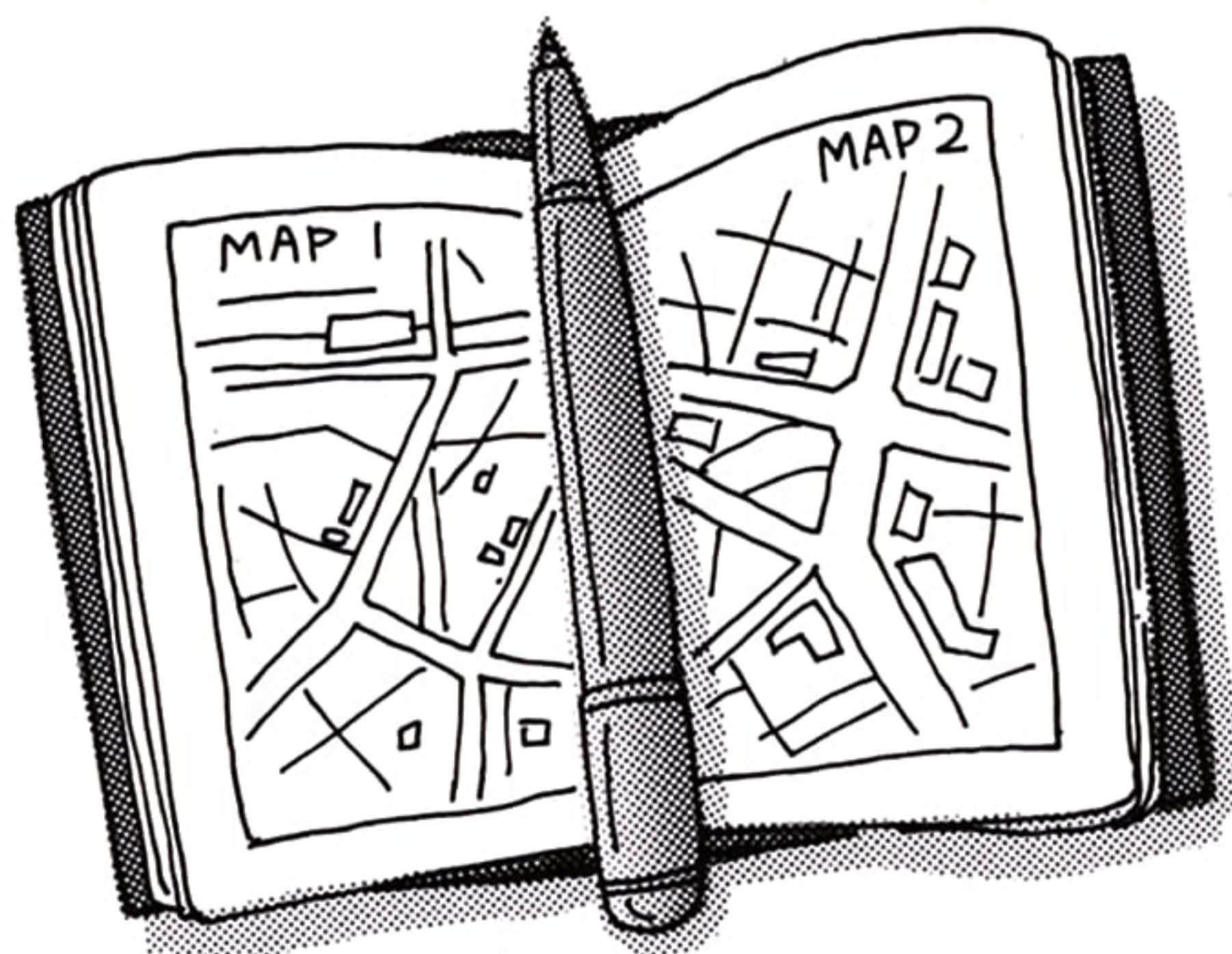
なお、DISKCOPYコマンドは内蔵コマンドではなく外部コマンドです。実行するためにはプログラムファイル「DISKCOPY.EXE」(あるいは.COM)がディスク上に必要です。

```
A>DISKCOPY A: B:/V⏎ ……カレントドライブ(A:)上のDISKCOPYコマンドを実行して、
                        ドライブA:とドライブB:のディスクを比較する

DISKCOPY version 4.00

ディスクの照合を行います

送り側ディスクをドライブ A: に挿入してください
受け側ディスクをドライブ B: に挿入してください
準備ができたらどれかのキーを押してくださいx ……適当なキーを入力する

照合は終了しました ……2つのディスクの内容は完全に一致していた
もう一度実行しますか(Y/N)Y ……ドライブB:を別のディスクと入れ換えて、再度実行する

DISKCOPY version 4.00

ディスクの照合を行います

送り側ディスクをドライブ A: に挿入してください
受け側ディスクをドライブ B: に挿入してください
準備ができたらどれかのキーを押してくださいx

トラック  0 の内容が異なっています
中止 <A> ,強行 <I> ? I ……「強行」というより、「次のトラックを比較するか?」という意味である

トラック  8 の内容が異なっています
中止 <A> ,強行 <I> ? A ……中止(アボート)する

照合は失敗しました ……「失敗」という表現は適切でない。この意味は「一致していない」ということである
もう一度実行しますか(Y/N)N

A>
```

図 7.18 DISKCOPY コマンドによる2つのディスクの比較の実行例

7.12 2つのファイルの比較 FC（ファイルコンペア）

― 外部コマンド、ワイルドカード使用不可 ―

ディスク上の2つのファイルの内容が同じかどうかは、FCコマンド(File Compare)を使って比較することができます。比較するファイルの形式は、文字ファイルでもプログラムファイル(マシン語ファイル)でも、どんなファイルでもかまいません。また、前節のDISKCOPYコマンドによるディスク全体の比較と異なり、FCコマンドはファイル単位の比較ですので、ディスクの種類にはまったく関係しません。「DISKCOPY/V」と、このFCコマンドによる比較の違いを明確に区別してください。

このFCコマンドには、比較の仕方に関するいろいろな機能がありますが、ここでの実習は、完全に一致するか／しないかだけの、最も単純な比較を行います。

FCコマンドの書式を次に示します。「*x*：」「*y*：」はドライブ名で、カレントドライブの場合は省略できます。

```
A>FC  x：ファイル名1  y：ファイル名2↵
```

```
A>FC/B  x：ファイル名1  y：ファイル名2↵
```

1つ目の書式は、文字ファイルの比較をするもので、バイナリファイル(マシン語ファイルなど)の比較をする場合には、2つ目の書式のように、/Bスイッチを付ける必要があります。

では実行例を示しましょう(図7.19)。FCコマンドは、内蔵コマンドではなく外部コマンドです。実行するためにはプログラムファイル「FC.EXE」(あるいは.COM)がディスク上に必要です。

```
A>FC/B COMMAND.COM B:COMMAND.COM↵
FC: 違いは見つかりません.

A>
```

……カレントドライブ(A:)上のマシン語ファイル「COMMAND.COM」と、B:上の同名のファイルと比較する。バイナリファイルの比較なので、/Bスイッチを指定してある

……両者は完全に一致した

― 図7.19 ― (次ページに続く)

```
A>FC/B B:CHAP1 B:CHAP1.BAK
00000147: 82 81
00000148: C9 69
00000149: 88 D9
0000014B: 91 DE
0000014C: B6 81
0000014D: 82 46
0000014E: B5 20
～～～～～～～～～～～～～
00003716: 6A 5C
00003717: 81 82
00003718: 42 C9
00003719: 0D 82
0000371A: 0A C8
0000371B: 1A 82
FC: B:CHAP1.BAK は B:CHAP1 よりサイズが大きいファイルです.

A>
```

カレントドライブ(A:)上の FC プログラムを実行して、B:上のファイル「CHAP1」と、同じくB:上の「CHAP1.BAK」とを比較する。2つのファイルはいずれも文字ファイルであるが、/B スイッチを付けて、バイナリファイルの比較形式(1バイトごとのデータを比較)で実行する

ファイルの頭から147(16進数)バイト目が一致しない。ファイル1が「82」であり、ファイル2は「81」である。(いずれも16進数)このように一致しない箇所がすべて表示される

```
A>B:
B>A:FC CHAP1 CHAP1.BAK
***** CHAP1
SIC( プログラミングの入門用プログラミング言語の1つ) の世界とは無縁であり、そこは
MS-DOSで代表されるOSを基盤とする世界です。
　ソフトウェア開発の専門家は当然のことながら、多くのビジネスソフトユーザーにとっ
ても、OSの知識は非常に重要です。コンピュータを
自由に活用するには、そのOSの基礎知識がぜひとも必要になるでしょう。
　本章では、MS-DOSというOSが何をする場合にも「基本ソフトウェア」として存在するこ
とを認識するとともに、なぜ必要なのかを解説しま
す。
　MS-DOS－－つまりOSとは何かということを、今すぐ理解する必要はありません。そう簡
単に説明や理解ができるものではないのです。しか
 ***** CHAP1.BAK
SIC( プログラミングの入門用プログラミング言語の1つ) の世界とは無縁であり、そこは
MS-DOSで代表されるOSを基盤とする世界です。
　ソフトウェア開発の専門家は当然のことながら、多くのビジネスソフト利用者にとって
も、OSの知識は非常に重要です。コンピュータを自
由に活用するには、そのOSの基礎知識がぜひとも必要になるでしょう。
　本章では、MS-DOSというOSが何をする場合にも「基本ソフトウェア」として存在するこ
とを認識するとともに、OSがなぜ必要なのかを解説
します。
　MS-DOS－－つまりOSとは何かということを、今すぐ理解する必要はありません。そう簡
単に説明や理解ができるものではないのです。しか
 *****

B>
```

カレントドライブをB:に移す。(とくに意味はない)

ドライブA:上のFCコマンドを実行して、さきほどと同じカレントドライブ(B:)上の文書ファイル「CHAP1」と「CHAP1.BAK」を比較する。今回は/Bスイッチがないので文字ファイルの比較形式で実行される

CHAP1の内容

CHAP1.BAKの内容

それぞれ▬▬の部分が異なる部分
このように/Bスイッチを付けない場合は、一致しない箇所を含む行が、それぞれの行単位で表示される。リターンからリターンまでが1行であり、画面上の1行とは異なる

図 7.19　FC コマンドの実行例

7.13 ファイル内の文字列サーチ FIND（ファインド）

― 外部コマンド、ワイルドカード使用不可 ―

FINDコマンドは、1つ、または複数のファイルの内容をサーチ(検索)して、指定した任意の文字列を捜し出します。これは、各種のエディタや、ワープロに備わっている「検索」(文字列サーチ)の機能と同じです。このFINDコマンドは、1つのファイルの内容をサーチするだけでなく、1度の実行で複数のファイルをまとめてサーチすることができます。指定した文字列が見つかれば、そのファイル名と、目的の文字列が含まれるすべての行を表示します。

またFINDコマンドは、12章で解説する**フィルタ**としても機能し、ほかのコマンドと組み合わせて使うこともできます。また、コマンドにスイッチを付けることによるいくつかの付加機能がありますが、ここでは最も基本的な使い方を実習します。

では実行例を示しましょう(図7.20)。FINDコマンドは、内蔵コマンドではなく外部コマンドです。実行するためにはプログラムファイル「FIND.EXE」(あるいは.COM)がディスク上に必要です。

```
A>FIND "コマンド・プロセッサ" B:CHAP1 B:CHAP2 B:CHAP3 ↵
----------- B:CHAP1

----------- B:CHAP2
　いずれにしろ、「DIR」というコマンドによって、画面には、そのディスクの内容が表示
されたと思います。このときMS-DOSのコマンド・プロセッサ(入力された命令(ﾙﾋﾞ:ｺﾏﾝﾄ
ﾞ)の識別実行処理部)と呼ばれる部分が、キー入力された「D」「I」「R」という文字列の
意味を識別し、そのコマンドのプログラム(MS-DOSに内蔵されている)を実行したわけで
す。

----------- B:CHAP3
というファイルが1つだけコピーされています。実はMS-DOS自身も、このディスク上に2つ
のファイルとして存在しているのですが、それらのファイルはDIRコマンドでは表示されな
い特別なファイルです。COMMAND.COMは、普通のプログラムファイルと同じ形式のファイル
ですが、MS-DOSシステムの一部分を構成する重要なプログラムであり、「コマンド・プロ
セッサ」と呼ばれています。このファイルがディスク上にない場合、MS-DOSはプロンプト
「A>」を表示して(コマンドレベルの状態で)立ち上がることができません。つまり、M
S-DOSを起動できる最も基本的で最小限の(いわば「裸の」)システムディスクは、次のよ
うなものになります。

A>
```

ドライブB:上の3つのファイルの中から検索する
「" "」で囲まれた文字列「コマンド・プロセッサ」を検索する
ドライブB:のファイル「CHAP1」には該当する文字列はなかった
ファイル「CHAP2」には該当する文字列があった。その行が表示されている
ファイル「CHAP3」にも該当する文字列があった。その行が表示されている
これらはそれぞれ復帰・改行がはいっていない1行の文。つまりディスプレイ表示上の1行とは異なる

図7.20　FINDコマンドの実行例

7.14 ディスクにディスク名を付ける /V スイッチ付 FORMAT（フォーマット） —外部コマンド—
ディスク名を調べる VOL（ボリューム） —内蔵コマンド—
ディスク名の付加、変更、削除 LABEL（ラベル） —外部コマンド—

7.14.1 ディスクにディスク名を付ける（/V スイッチ付 FORMAT）

ディスクのフォーマット処理（初期化）については、すでに 3.1、3.3 節で解説と実習を行いました。そこでも述べましたが、FORMAT コマンドには、次のスイッチによる付加機能があります。

/S …… フォーマット終了後、MS-DOS システムをコピーする
/V …… フォーマット終了後、ディスク名（ボリュームラベル）を付ける

ここでは/V スイッチ付 FORMAT コマンドを使って、個々のディスクに名前（**ボリュームラベル**という）を付ける機能の実習をしましょう（MS-DOS のバージョン 3.*x* には、このディスク名を付ける専用コマンド「LABEL」が用意されている。後述）。

ボリュームラベルは、DIR コマンドでも表示されますので、すでにお気付きの方もいると思いますが、個々のディスクに任意に付けることができる名前のことです。

ボリュームラベルは、どのような名前を付けてもかまいません。文字数の制限は、半角文字の場合は 11 文字まで、漢字など全角文字の場合は、1 文字が半角文字の 2 文字分ですので、5 文字まで使うことができます。半角／全角を混用することもできますが、文字数の計算に注意してください。つまり、文字数や使える文字の種類などの制限は、ファイル名の場合とまったく同じです（8 文字＋3 文字＝11 文字）。

では FORMAT コマンドを実行して、ボリュームラベルを付けてみましょう。/V スイッチ付 FORMAT コマンドの書式を示します。「*x*：」はドライブ名で、カレントドライブの場合は省略できます。

```
A>FORMAT x：/V ↵
```

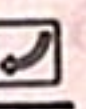
なおスイッチは、MS-DOS システムをコピーする/S スイッチと同時に、「/S/V」と指定することもできます。いずれの場合もディスクのフォーマット処理が行われ、ディスク上の内容は完全に消去されますので注意してください。

FORMAT コマンドは、内蔵コマンドではなく外部コマンドです。これを実行するには、プログラムファイル「FORMAT.EXE」（あるいは .COM）がディスク上に必要です。

では実行例を示します。ここではボリュームラベルを付けるとともに、/S スイッチも付けて、MS-DOS システムをコピーしていますが、MS-DOS システムをコピーする必要のない場合は、/S スイッチは不要です（図 7.21）。

```
A>FORMAT B:/S/V↵        ……FORMATコマンドで/Sスイッチ、/Vスイッチを指定し、
Format Version 4.10         ボリュームラベル付きのシステムディスクを作る


新しいディスクをドライブ B: に挿入し
どれかキーを押してください

ディスクのタイプは    1 : 640(KB)  2 : 1(MB)  = 2↵

目的のディスクは 1MB FD です


フォーマット中です. . .    残り  153 トラックです
0  10  20                    70  80  90  100
システムを転送しました

   1250304 バイト  全ディスク容量
    120832 バイト  システム領域
   1129472 バイト  使用可能ディスク容量


ボリュームラベルを入力してください
漢字<全角>は５文字、英数字<半角>は１１文字まで、必要なければリターンキー

ボリュームラベル＝SAMPLE↵ ……適当なボリュームラベルを入力する

別のディスクをフォーマットしますか(Y/N) N↵
A>DIR B:↵ ……作成したディスクのディレクトリを確認する

 ドライブ B: のディスクのボリュームラベルは SAMPLE ←入力したボリュームラベル
 ディレクトリは B:¥

COMMAND  COM     24931  88-07-13    0:00 ……MS-DOSシステムのみが存在している
         1 個のファイルがあります.
  1129472 バイトが使用可能です.

A>
```

図 7.21　/V スイッチ付 FORMAT コマンドによるディスク名の付加実行例

7.14.2 ディスク名を調べる(VOL)

次は、ディスク名(ボリュームラベル)を調べるコマンドVOLの使い方です。これは非常に簡単なコマンドなのでとくに解説する部分もありません。コマンドの書式を示します。

```
A>VOL x:⏎
```

「x：」は調べる対象のドライブ名ですが、それがカレントドライブの場合は省略できます。VOLコマンドは内蔵コマンドであり、実行するためのプログラムはMS-DOSに内蔵されていますので、ディスク上に存在する必要はありません。実行は、カレントドライブに関係なく行えます。では実行例を示します(図7.22)。

```
A>VOL⏎ ……………カレントドライブ(A:)上のディスクのボリュームラベルを調べる

 ドライブ A: のディスクのボリュームラベルはありません. …………システムディスクにはボリューム
                                                              ラベルは付いていない
A>VOL B:⏎ ……………ドライブB:上のディスクのボリュームラベルを調べる

 ドライブ B: のディスクのボリュームラベルは SAMPLE ……………さきほど作成したディスク

A>
```

図7.22 VOLコマンドの実行例

7.14.3 ディスク名の付加、変更、削除(LABEL)

MS-DOSのバージョン3.xから、ディスク名(ボリュームラベル)に関する専用コマンドLABELが提供されています。MS-DOSのバージョン2.xでは、ディスク名は、FORMATコマンドの/Vスイッチでなければ付けることができませんでしたが、このLABELコマンドを使えば、フォーマット処理に関係なく、使用中のディスクに対して自由にディスク名の付加、変更、削除することができます。ディスクの内容には、まったく影響はありません。

LABELコマンドの書式を示します。「x：」は目的のドライブ名で、カレントドライブの場合は省略できます。

```
A>LABEL x:⏎
```

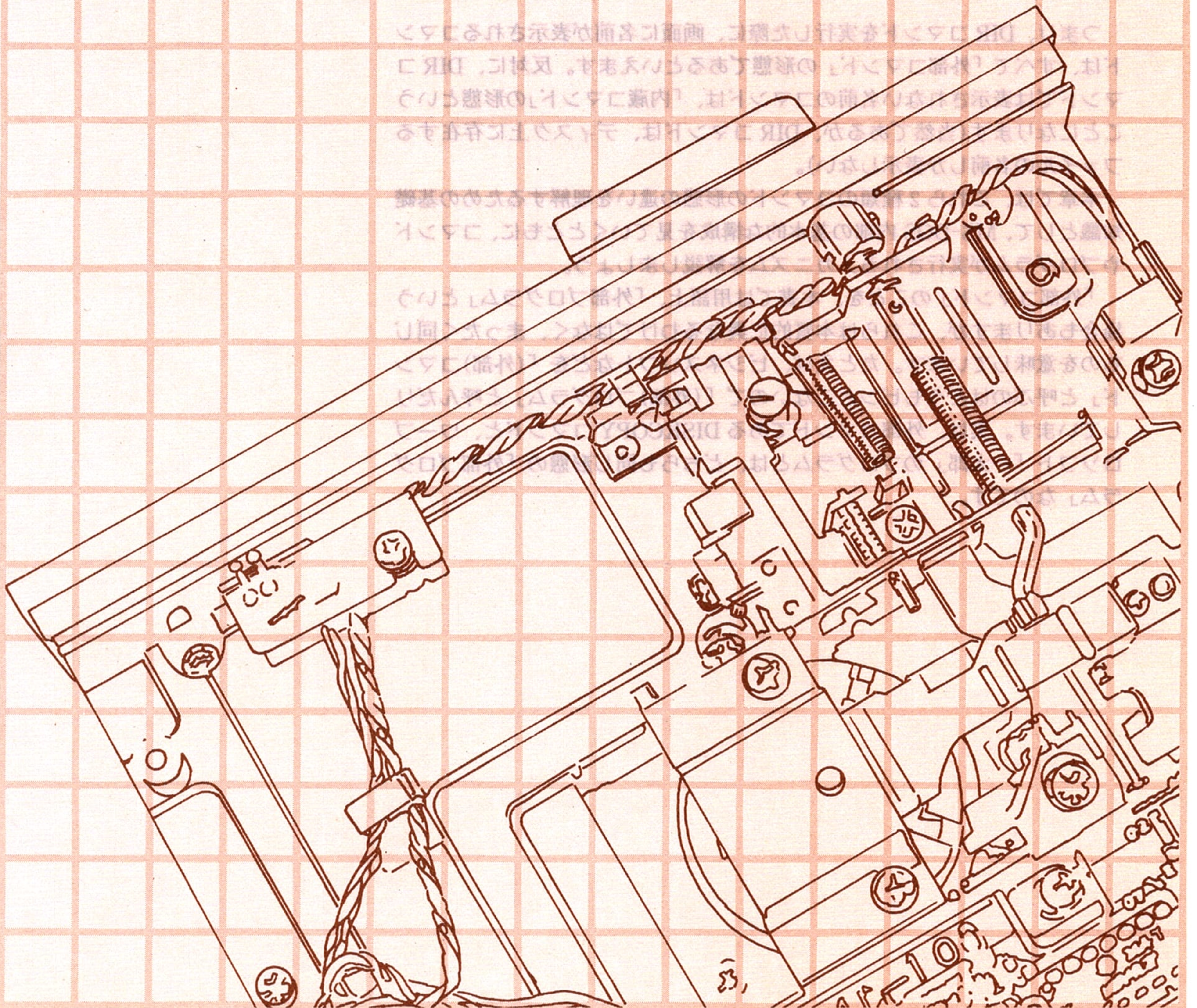
```
A>LABEL x:ディスク名⏎
```

１つ目の書式の場合は、表示されるメッセージに従って選択、操作することにより、ディスク名の付加、変更、削除ができますが、２つ目の書式の場合は、ドライブ*x*：のディスクに、指定した「ディスク名」がいきなり書き込まれます。すでにディスク名が付けられていた場合は、新しいディスク名に書き換えられますので注意してください。

では実行例を示しましょう。LABELコマンドは、内蔵コマンドではなく外部コマンドです。これを実行するには、プログラムファイル「LABEL.EXE」がディスク上に必要です。

```
A>LABEL B:↵ ……ドライブB:のディスクのボリュームラベルを変更する目的でLABELコマンドを実行する

ドライブ B のボリュームラベルは SAMPLE ……現在は「SAMPLE」になっている

ディスクのボリュームラベルを入力してください.
漢字<全角>は 5 文字，英数字<半角>は 11 文字まで
必要なければ <改行キー> : LABEL_CMD ……「LABEL_CMD」に変更する

A>VOL B:↵ ……変更したボリュームラベルを確認する

 ドライブ B: のディスクのボリュームラベルは LABEL_CMD ……変更されている

A>LABEL B:↵ ……ドライブB:のディスクのボリュームラベルを削除する目的でLABELコマンドを実行する

ドライブ B のボリュームラベルは LABEL_CMD

ディスクのボリュームラベルを入力してください.
漢字<全角>は 5 文字，英数字<半角>は 11 文字まで
必要なければ <改行キー> :↵ ……↵のみを入力する

現在のボリュームラベルを削除します.
よろしいですか <Y/N>? Y↵ ……「Y」を入力して、ボリュームラベルを削除する

A>VOL B:↵ ……確認する

 ドライブ B: のディスクのボリュームラベルはありません. ……ボリュームラベルは削除されている
A>
```

図7.15　LABELコマンドの実行例

8章
MS-DOSの構成と内蔵コマンドおよび外部コマンド

今までの実習からも、MS-DOS のコマンドには、その実行前のプログラムが存在する場所の違いによる、次の 2 つの形態があることはすでにお気付きのことと思います。

内蔵コマンド …… DIR、COPY、REN、DEL、TYPE、など
　そのコマンドのプログラムが MS-DOS の内部に存在する。「内部コマンド」「組み込みコマンド」などとも呼ばれる

外部コマンド …… FORMAT、DISKCOPY、SYS、CHKDSK、ATTRIB、LABEL など
　そのコマンドのプログラムがディスク上に存在する。「外部プログラム」などとも呼ばれる

つまり、DIR コマンドを実行した際に、画面に名前が表示されるコマンドは、すべて「外部コマンド」の形態であるといえます。反対に、DIR コマンドでは表示されない名前のコマンドは、「内蔵コマンド」の形態ということになります(当然であるが、DIR コマンドは、ディスク上に存在するファイルの名前しか表示しない)。

本章では、これら 2 種類のコマンドの形態の違いを理解するための基礎知識として、MS-DOS 内部の基本的な構成を見ていくとともに、コマンドやプログラムが実行されるメカニズムを解説しましょう。

「外部コマンド」のことを、本書では用語上、「外部プログラム」という場合もありますが、これらは本質的に異なるわけではなく、まったく同じものを意味しています。たとえば、ビジネスソフトなどを「(外部) コマンド」と呼ぶのはどうもピンとこないので「(外部) プログラム」と呼んだりしています。実は、外部コマンドである DISKCOPY コマンドと、ワープロソフト「一太郎」のプログラムとは、どちらも同じ形態の「外部プログラム」なのです。

8.1 MS-DOSの構成と内蔵コマンド

MS-DOSの代表的なコマンドである、DIR、COPY、REN、DEL、TYPEなどを実行するためのプログラムは、MS-DOS自身の**コマンド・プロセッサ**(コマンド処理部。ディスク上のCOMMAND.COMが読み込まれた部分)と呼ばれる部分に組み込まれています。

MS-DOS自身は、図8.1に示すように3つの部分から構成されており、起動時にディスク(MS-DOSのシステムディスク)から読み出されて、このようにコンピュータのメモリ上の所定の位置に配置されます。

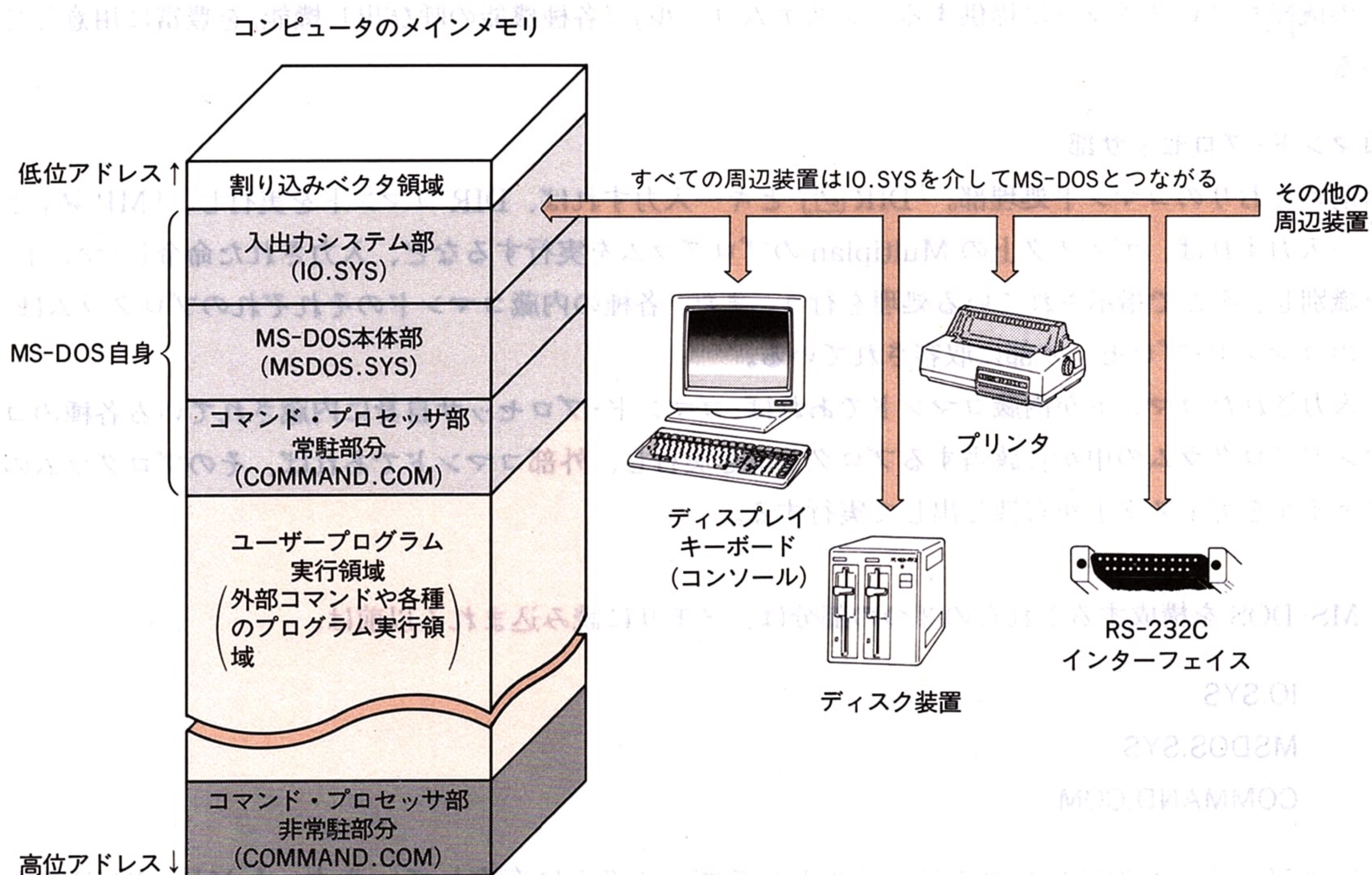
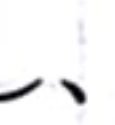



図8.1　起動しているMS-DOSのメモリ上の構成

MS-DOSを構成する3つの部分は、それぞれ次のような働きをします。

入出力システム部

MS-DOSの本体部(中心核)と、コンピュータのハードウェア(キーボード、ディスプレイ、ディスク装置、プリンタ等の各種周辺装置など)とをソフトウェア的に結びつけるための部分。各周辺装置等のハードウェアの内部は、それぞれのメーカーによって、ハードウェア的にも、ソフトウェア的にも異なっている。したがって、MS-DOSの中でこの部分だけは、各メーカーの責任において、自社の機種上でMS-DOSがうまく動作するように作成しなければならない。この部分の作り方次第で、同じMS-DOSでも性能に差が出る。

MS-DOS本体部

MS-DOSの心臓部。階層構造(ツリー構造)を持つディレクトリや、FAT(ファイルアロケーション・テーブル)による高速ディスクアクセス(読み／書き)を可能にする、優れたファイル管理システムを実現する部分。さらに、MS-DOS上でのソフトウェア開発をサポートするために、MS-DOS内部の多くの機能をプログラマーに提供する「システムコール」(各種機能の呼び出し機能)を豊富に用意している。

コマンド・プロセッサ部

名前のとおりのコマンド処理部。「DIR⏎」とキー入力すれば、DIRコマンドを実行し、「MP⏎」とキー入力すれば、ディスク上のMultiplanのプログラムを実行するなど、入力された命令(コマンド)を識別し、そこで指示されている処理を行う。また、各種の内蔵コマンドのそれぞれのプログラムは、このコマンド・プロセッサ部に収容されている。

　入力されたコマンドが内蔵コマンドであれば、コマンド・プロセッサ自身に内蔵されている各種のコマンドプログラムの中から該当するプログラムを実行し、外部コマンドであれば、そのプログラムのファイルをディスク上から捜し出して実行する。

　MS-DOSを構成するこれらの3つの部分は、メモリに読み込まれる以前は、

IO.SYS
MSDOS.SYS
COMMAND.COM

というファイル名のプログラムファイルとしてディスク上に存在しています。ただし、IO.SYSとMSDOS.SYSの2つのファイルは、DIRコマンドでは表示されない特別のファイルです(11章の図11.13のCHKDSKコマンドの実行例に、その姿を見ることができる)。しかしCOMMAND.COMは、一般のファイルと同じ通常の形態のファイルであり、DIRコマンドの表示でもおなじみのものです。

さて、さきに述べたように、MS-DOS の内蔵コマンドを実行するための個々のプログラムは、メモリ上の MS-DOS システムのコマンド・プロセッサ部にすべて内蔵されています(コマンド・プロセッサ部は、ディスク上のファイル「COMMAND.COM」が MS-DOS の起動時にメモリに読み込まれたもの)。つまり、MS-DOS が起動して、「A>」や「B>」などが表示されているコマンドレベルでは、内蔵コマンドのプログラムは、すべてがメモリ上に用意されており、いつでも実行可能な状態にあるわけです。

ですから、プロンプトのあとに、何らかの内蔵コマンドのコマンド名が入力され⏎されると、コマンド・プロセッサの命令解析部がその文字列を判別し、ディスク上のプログラムファイルを捜すことなく、直ちにメモリ上に用意されている該当プログラムを実行します。このように内蔵コマンドは、ディスク上にそれぞれのプログラムファイルを用意する必要がなく、したがって「カレントドライブ」を気にする必要もなく、いつでも即実行できる便利なものです。(図 8.2)

②コマンド・プロセッサ部に「DIR ⏎」のコマンドラインが送られる
⑤DIRコマンドによる結果の表示が行われる
入出力システム部
(IO.SYS)
MS-DOS 本体部 (MSDOS.SYS)
①「DIR ⏎」とキー入力する
ユーザープログラム実行領域 (外部コマンドや各種プログラム実行領域)
③入力されたコマンドは、内蔵コマンドのDIRであることが判別され、該当プログラムが実行される
命令判別実行指示部
DATE、TIMEプログラム
DIRプログラム
RENプログラム
DELプログラム
TYPEプログラム
その他内蔵プログラムの全プログラム
コマンド・プロセッサ部 (COMMAND.COM)
④DIRコマンドのプログラムをMS-DOSが実行し、その結果が得られる

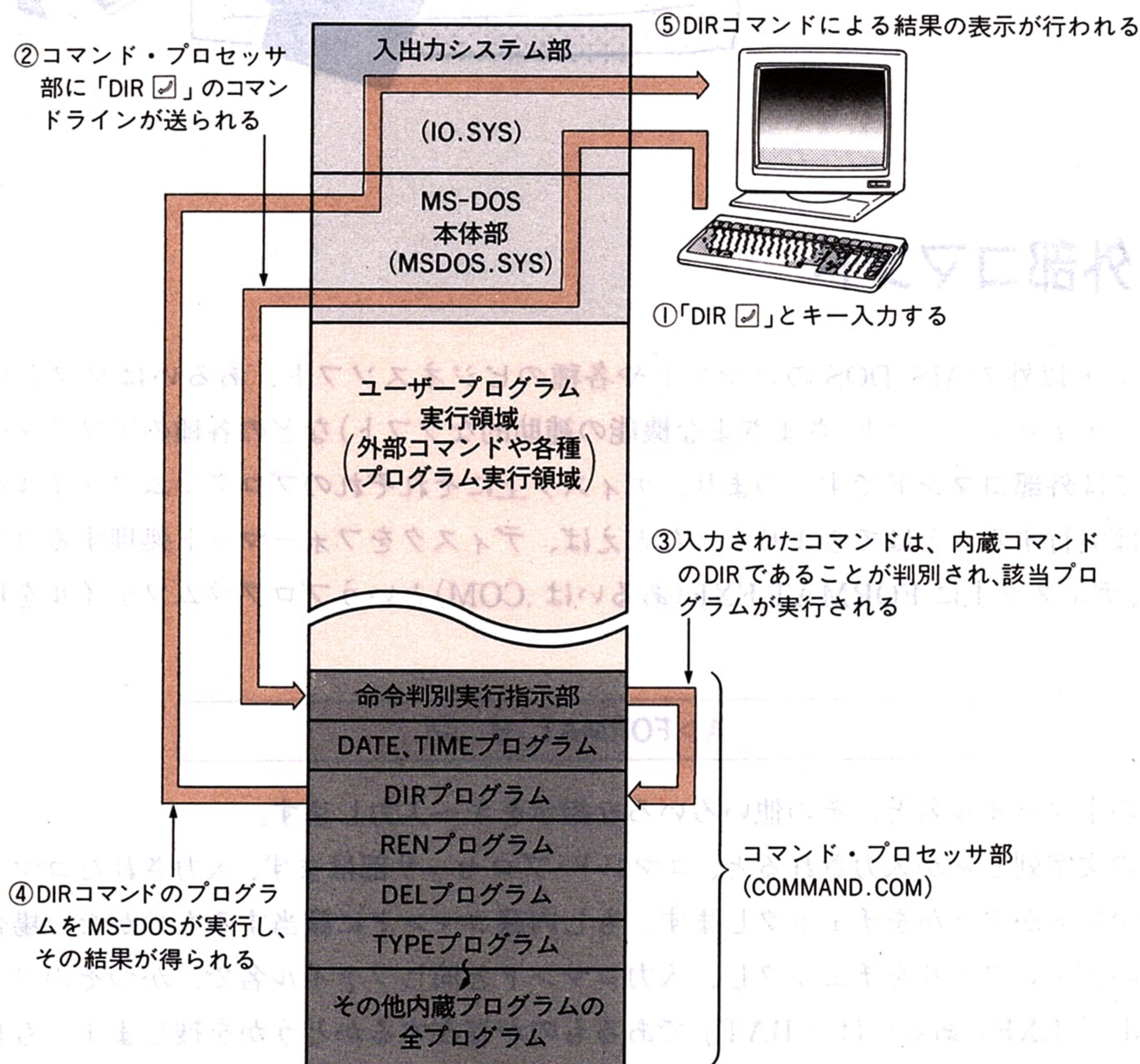
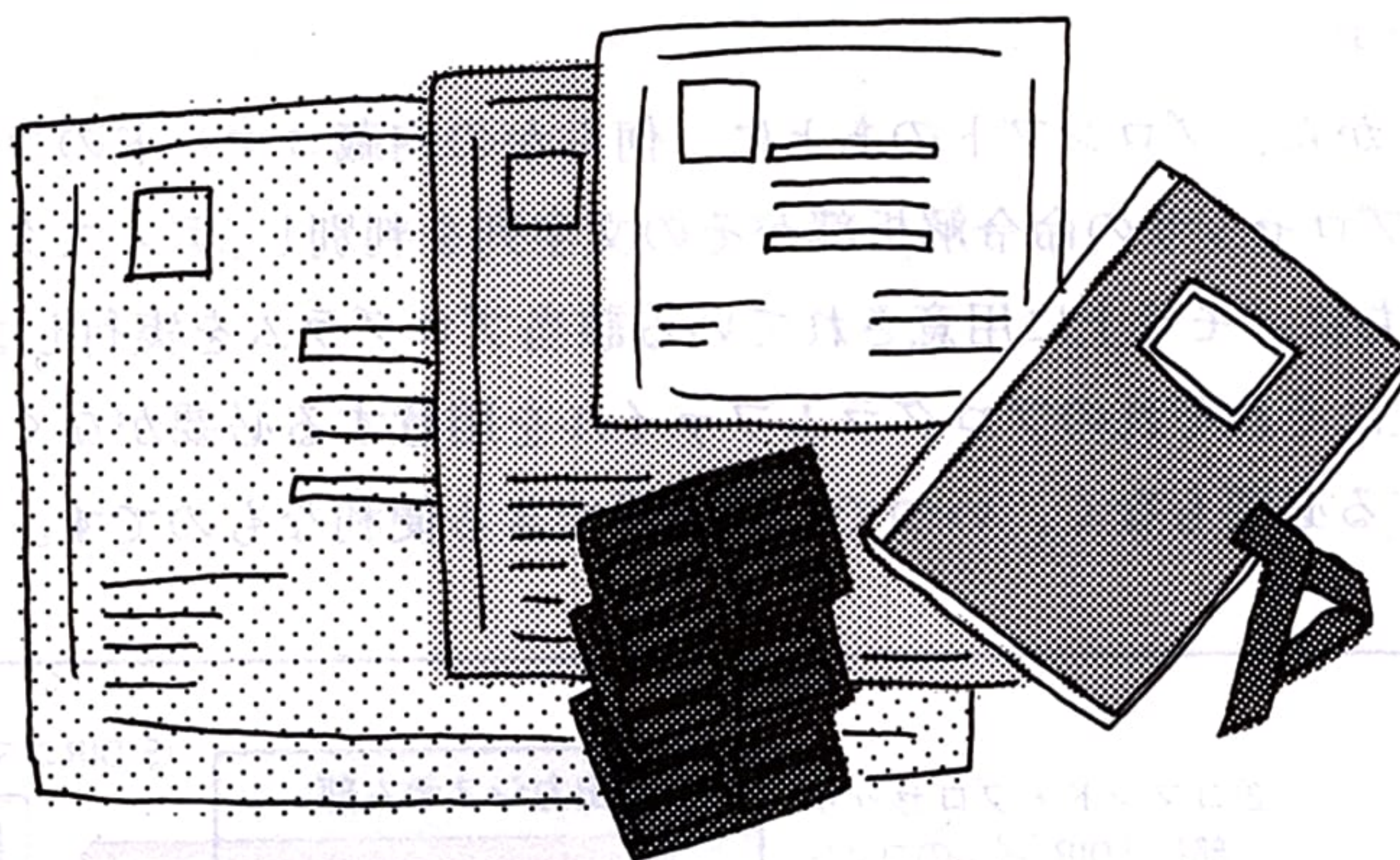
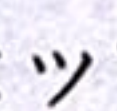



図 8.2　コマンド・プロセッサの働きと内蔵コマンド実行の流れ

しかし内蔵コマンドは、メモリ領域(プログラムの大きさ)の制限や、コマンド・プロセッサというMS-DOSシステムの共通部分に内蔵する関係から、内蔵可能なプログラム(コマンド)の種類は、コンピュータの機種に依存しない、なるべくコンパクトなもので、かつ日常多用するコマンドのいくつかに限られます。

8.2　外部コマンド

内蔵コマンド以外のMS-DOSのコマンドや各種のビジネスソフト、あるいはソフトウェア開発ツールやユーティリティソフト(さまざまな機能の補助的なソフト)などの各種のアプリケーションソフトのすべては**外部コマンド**です。つまり、ディスク上にそれぞれのプログラムファイルを用意しておかなければ実行することはできません。たとえば、ディスクをフォーマット処理するコマンドを実行するには、ディスク上にFORMAT.EXE(あるいは.COM)というプログラムファイルを用意した上で、

```
A>FORMAT B:⏎
```

などと、その主ファイル名と、その他いろいろな指示をキー入力します。

コマンドの文字列と⏎が入力されると、コマンド・プロセッサ部はまず、入力されたコマンドの文字列が内蔵コマンドかどうかをチェックします。もし内蔵コマンドに該当するものがない場合は、次にディスク上のディレクトリをチェックし、入力コマンドと同じファイル名で、かつそのファイルタイプが「.COM」「.EXE」あるいは「.BAT」であるものが存在するかどうかを捜します。もし該当するものが存在すれば、そのファイル(プログラムファイル)をコンピュータ本体のメモリに読み込み実行します。その動作の流れを**図8.3**に示しましょう。

入出力システム部
(IO.SYS)
MS-DOS
本体部
(MSDOS.SYS)
コマンド・プロセッサ常駐部
FORMATプログラム
ユーザープログラム
実行領域
(外部コマンドや各種
プログラム実行領域)
命令判別実行指示部
DATE、TIMEプログラム
DIRプログラム
RENプログラム
DELプログラム
TYPEプログラム
その他内蔵プログラムの
全プログラム
コマンド・プロセッサ部
(COMMAND.COM)
①「FORMAT B:⏎」と
キー入力する
②コマンド・プロセ
サ部に、
「FORMAT B:⏎」
のコマンドライン
が送られる
③FORMATコマンドは内蔵コ
マンドではないので、ディ
スク上を捜す
④該当ファイルが存在してい
たので、そのプログラム
(FORMAT.EXE)をメモリに
読み込む
⑤FORMATコマンドを、
MS-DOSが実行する
⑥ドライブB:のディスクがフ
ォーマット処理される

図 8.3　外部コマンドの実行の流れ

実行可能な外部コマンドのファイル(つまりマシン語のプログラムファイル)の名前には、必ず次の2種類のどちらかのファイルタイプが付きます。

主ファイル名 . COM
主ファイル名 . EXE

実行可能なマシン語ファイルには、必ずこれらのファイルタイプが付いていますが、これらの違いについては6.3節で触れましたので参照してください(表6.3)。この両者は、プログラムサイズの違いや、プログラムがメモリ上に読み込まれる際の状況が少し異なりますが、一般のユーザーにとって、

この両者の違いを意識する必要はありません。両者とも同じ「実行可能なマシン語ファイル」と考えてけっこうです。

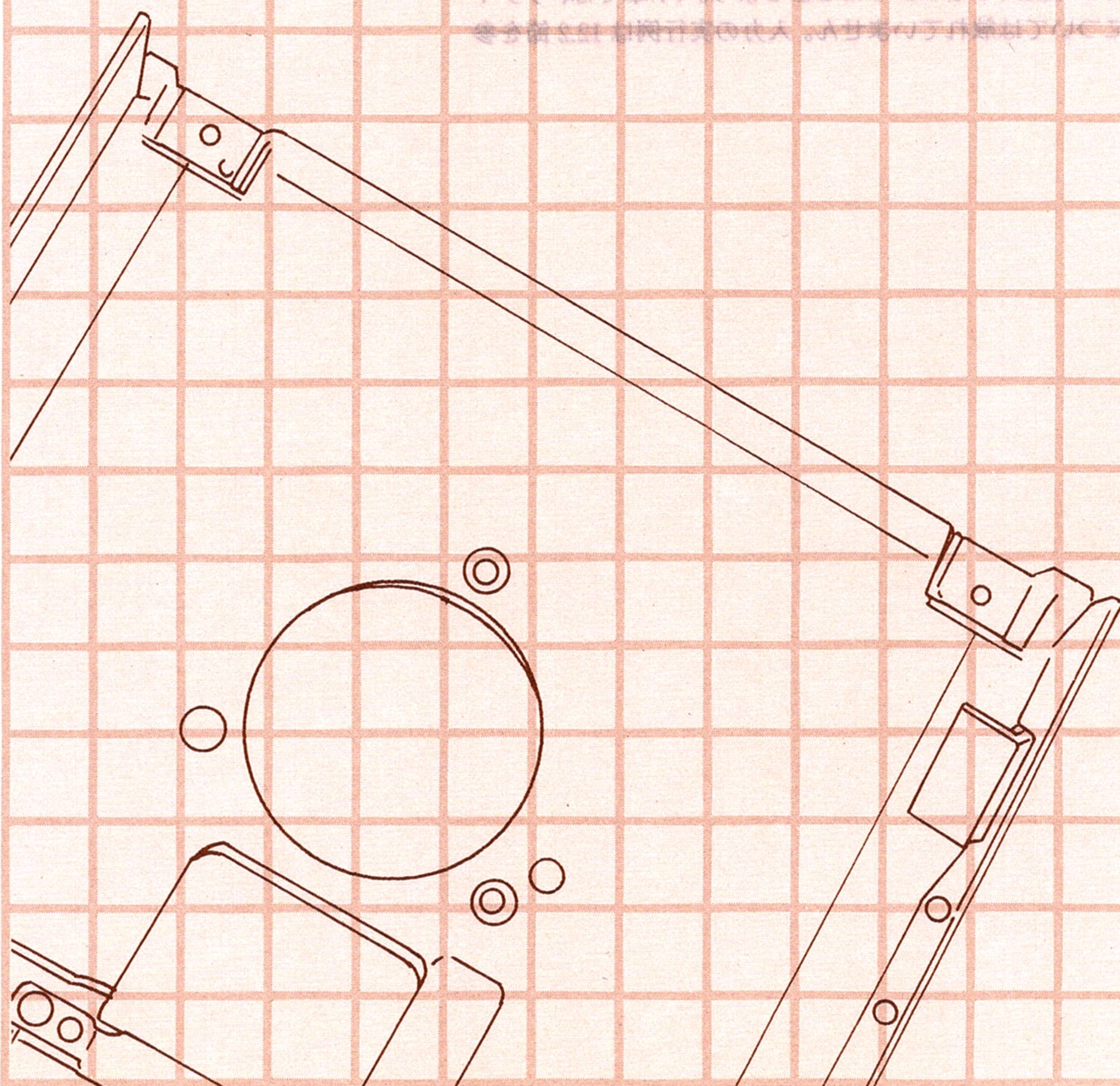
手もとにあるディスク上のファイルを、DIRコマンドで、

```
A>DIR *.COM↵
```

```
A>DIR *.EXE↵
```

のように実行して、実行可能なディスク上のプログラム、つまり外部コマンドを確認しておいてください。

なおもう1つ、「.BAT」のファイルタイプが付いた**バッチファイル**と呼ばれるファイルも実行が可能です。これについては10章で解説します。

9章
周辺装置間のデータのやりとり

7.6 節において、COPY コマンドは「周辺装置間のデータ転送コマンド」であることを解説しました。しかし、ある周辺装置から、ある周辺装置へ、データを転送することができるのは、COPY コマンドだけではありません。たとえば 5.3 節で解説した、ディスプレイへの出力をプリンタへも同時に出力させる CTRL + P の機能も、データ転送の 1 つの手段ですし、また本章で新たに登場する「リダイレクト」という機能も、データ転送の重要な手段です。

本章では、データ転送を考える上での基礎知識となる、「デバイスファイル」の概念(各周辺装置もファイルであるという概念)を解説するとともに、リダイレクトによるデータ転送を行うための「リダイレクト記号」(入力／出力転送記号) ——「<」および「>」—— の使い方などを実習します。このリダイレクトによるデータ転送は、結果的には COPY コマンドと同じような動作に見えますが、概念的にはまったく性質の異なる転送ですので、その違いをよく理解して混同しないようにしましょう。(本章では、リダイレクトによる入力については触れていません。入力の実行例は 12.2 節を参照)

9.1 周辺装置もファイル

MS-DOS上で、COPYコマンドや、リダイレクト(後述)を使って周辺装置を呼ぶ(指定する)場合、それらは次に示す特別の名前で呼ばれます。

キーボードおよびディスプレイ	**CON(コンソール)**
プリンタ	**PRN(プリンタ)**
その他(一般にはRS-232Cインターフェイス)	**AUX(オグジャリ)**
ダミー装置(実態のない仮想の装置)	**NUL(ヌル)**

これらの周辺装置は**デバイスファイル**と呼ばれ、ディスク上のファイルと同じような概念で取り扱うことができます。このことから、**CON**、**PRN**、**AUX**などの名前は、**予約ファイル名**(すでにMS-DOSによって使われている名前)であり、私たちがディスク上のファイル名に使ってはいけないことになっています。6.2節の「ファイル名」において、使用禁止のファイル名が存在するのは、このためなのです。

次に、これらの装置が入力装置であるか出力装置であるかを考えてみましょう(**NUL**というダミー(仮想)のデバイスファイルについては、ここでは触れない)。

CON ⋯ 入力(キーボード)
　　　 ⋯ 出力(ディスプレイ)
PRN ⋯⋯ 出力(プリンタの印字)
AUX ⋯ 入力(たとえばRS-232Cインターフェイスの入力側)
　　　 ⋯ 出力(たとえばRS-232Cインターフェイスの出力側)

デバイスファイルを取り扱う場合には、この入力、出力の関係に注意してください。たとえば、「CON」という1つのデバイスファイルが、場合によってはキーボードであったり、ディスプレイであったり、2種類の装置を表します。

このような、それぞれのデバイスファイルに対する実際の装置の割り当て(たとえば、コンソール「CON」が、キーボードとディスプレイであることなど)は、ASSIGNコマンドにより、その他の装置に変更することが可能です(このコマンドは本書では取り扱わない)。上に示してあるそれぞれの割り当ては、初期状態(**イニシャルセット**とか**デフォルト**とか呼ばれている状態)のままのもので、MS-DOSが起動したときに標準的にセットされるものです。

次に、7.6節のCOPYコマンドのところで示した図(図7.8)をもとに、各種の周辺装置間のデータの流れを示しましょう(**図9.1**)。実際の周辺装置と、デバイスファイル名との対応に注目してください。

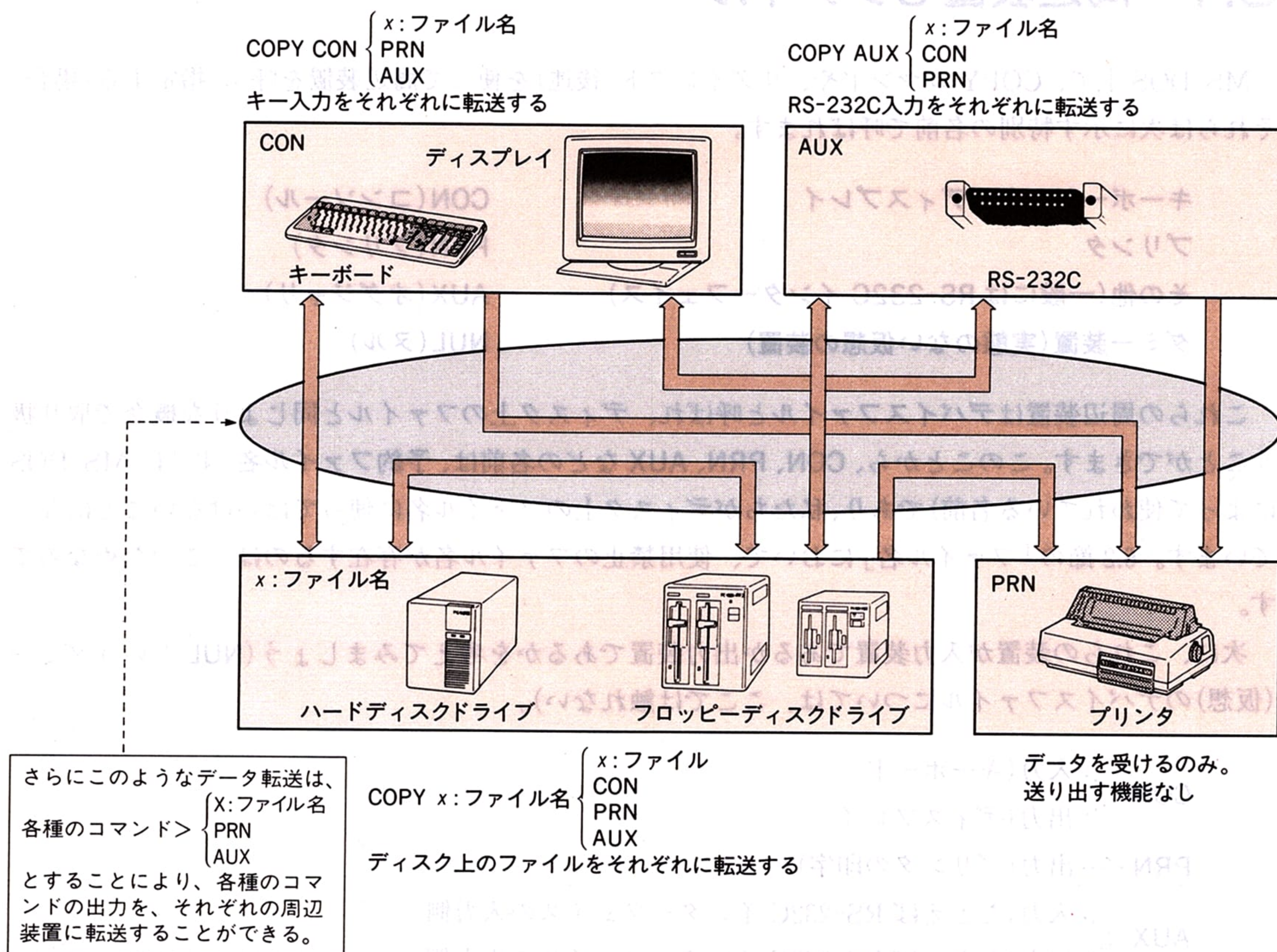



図 9.1　各種周辺装置間のデータの流れ

これらのデバイスファイル名を使って、COPY コマンドやリダイレクトにより、各装置間のデータ転送をディスク上のファイルと同じように行うことができます。次の節ではこの図をもとに、COPY コマンドと、リダイレクトを実習解説しましょう。

9.2 COPY コマンドによる周辺装置間のデータ転送

COPY コマンドについては、7.6 節ですでに解説しています。その中で、COPY コマンドは、ディスク上のファイルをディスク上にコピーするだけでなく、周辺装置間のデータ転送プログラムとして捉えなければならないことを解説し、実行例も示しました。もし COPY コマンドの理解が十分でない方は、もう一度 7.6 節を参照してください。

COPY コマンドの実習は、デバイスファイル名を使ったものも含めて、今までに 7.6 節などで次の 3 種類のことを行っています。

COPY　x:ファイル名 1　y:ファイル名 2 …… ドライブ x：上のファイル 1 を、ドライブ y：上にファイル 2 の名前でコピーする

COPY　CON　x:ファイル名 …… キーボード入力による文字ファイルを、ドライブ x：上に作成する

COPY　x:ファイル名　PRN …… ドライブ x：上の文字をファイルを、プリンタに出力する

ここでは、これらの実習を再度重複しては行いませんが、「周辺装置もファイルである」という認識のもとに、7.6 節での実習が、図 9.1 のデータの流れのどれに当るかを見ておいてください。

次に、まだ実習していない COPY コマンドによる周辺装置間のデータ転送の例を 2 つ示しましょう。

a) キー入力をディスプレイに出力する例

b) キー入力をプリンタに出力する例

「デバイスファイル間のデータ転送」ということに注目して実習してみましょう (図 9.2、図 9.3)。

```
A>COPY CON CON ⏎ ……コンソール入力(キー入力)をコンソール(ディスプレイ)へ出力するためのCOPYコマンドの実行
ABCDEFGHIJKLMN ⏎                        ┐
キーボードからディスプレイにコピーする ⏎   │ 適当に入力していく
1234567890 ⏎                            │
ｵﾜﾘ ⏎                                   ┘
^Z ⏎ ……CTRL+Z を入力し、COPYコマンドを終了する
ABCDEFGHIJKLMN
キーボードからディスプレイにコピーする ……「CTRL+Z ⏎」の入力と同時に、これが表示さ
1234567890                               れる。つまりCONからの入力がCONに出力された
ｵﾜﾘ
        1 個のファイルをコピーしました. ……COPYコマンド終了のメッセージ。キー入力したものを
                                          CONへ送ったという意味
A>
```

図 9.2　CON(コンソール)から CON(コンソール)へのコピー

```
A>COPY CON PRN ↵ ……………コンソール入力(キー入力)をプリンタに出力するためのCOPYコマンドの実行
ABCDEFGHIJKLMN ↵
abcdefghijklmn ↵
ｱｲｳｴｵｶｷｸｹｺ ↵
入門ＭＳ－ＤＯＳ　「デバイス・ファイル名」↵
↵ ……………ただの改行
おしまい。↵
^Z ↵ ……………CTRL+Z を入力し、COPYコマンドを終了する
          1 個のファイルをコピーしました.……………つまり、キー入力したものをPRNへ送ったという意味
A>
```

適当にキー入力していく。各行の終わりには必ず↵を入力すること。

この CTRL+Z を入力し、↵すると同時に今まで入力したデータが「PRN」へ送られ、印字が開始される

↓

最後の CTRL+Z ↵の入力により、以下の印字が開始される

```
ABCDEFGHIJKLMN
abcdefghijklmn
ｱｲｳｴｵｶｷｸｹｺ
入門ＭＳ－ＤＯＳ　「デバイス・ファイル名」

おしまい。
```

図 9.3　CON(コンソール)から PRN(プリンタ)へのコピー

■ RS-232C インターフェイスについて

コンピュータとコンピュータとの通信を行ったり、コンピュータと外部機器とのデータのやり取りを行ったりる場合に、多くのパーソナル・コンピュータは、RS-232C インターフェイスを利用します。この RS-232C インターフェイスは、たいていの機種で **AUX** という名前のデバイスファイルに割り当てられています。RS-232C を介してのデータの入出力は、COPY コマンドを使うと一部不都合な点があるため、専用コマンドを別途提供しているメーカーもあります(たとえば、NEC の PC-9800 用の MS-DOS には、「COPYA」という外部コマンドが用意されている)。

これらのコマンドを使えば、今までの実行例と同様に、RS-232C インターフェイスを介してのデータ転送を行うことができます。ただし、RS-232C インターフェイスを使うにあたっては、その初期設定とともに、ボーレート(通信速度)や、やり取りするデータのビット数などをはじめ、いくつかの設定条件を、送受双方で正しく合わせておかないと動作しませんので注意してください。

9.3 リダイレクトによるコマンド出力の転送

各種のコマンドによる出力(DIR コマンドによる表示など)を、リダイレクト記号「>」を使って各種の「ファイル」に転送することができます(「>」の前後のスペースは入れても入れなくてもよい)。

さっそく実行例を示しましょう。まず、DIR コマンドの表示出力を、デバイスファイル「PRN」(プリンタ)に出力します。つまり、DIR コマンドの表示内容をプリントアウトするわけです(CTRL + P をONにしても結果的には同じようにプリントアウトできるが、概念的には、まったく異なる)(図 9.4)。

```
A>DIR/W > PRN ⏎ ……………DIRコマンドのワイド表示の出力を、リダイレクトを使ってPRN(プリンタ)へ出力する

A> ……………ディスプレイには何も表示されない。つまりDIRコマンドの出力はPRNにつながっている
```

⬇

通常はディスプレイに表示されるものが、このようにプリンタに印字される

```
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COMCONFIG  SYSATOK6A  SYSATOK6B  SYSATOK     DIC
PRINT    SYSRSDRV   SYSMOUSE   SYSFORMAT  EXEDISKCOPY EXE
CHKDSK   EXESYS     EXEFC      EXEFIND    EXEMORE     COM
SORT     EXETREE    EXEATTRIB  EXEEDLIN   EXEREADME   DOC
        20 個のファイルがあります.
    358400 バイトが使用可能です.
```

図 9.4 リダイレクト「>」を使って、DIR コマンドの表示出力をプリントアウトする

次に、図 9.4 と同じ DIR コマンドの表示出力を、今度はディスク上のファイルに出力します。つまり、DIR コマンドの表示出力を内容とした新しいファイルを作成するわけです。そのファイル名を「DISKDATA.001」としましょう(図 9.5)。

```
A>DIR/W >DISKDATA.001↵ ……DIRコマンドのワイド表示の出力を内容とするディスク上のファイルを
ディスプレイには何も表示されない          リダイレクトを使って作成する。そのファイル名を「DISKDATA.001」とする
A>DIR DISKDATA.001↵ ……作成されたファイルを確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

DISKDATA 001        428  89-07-30   13:28 ……確かに作成されている
        1 個のファイルがあります.
   357376 バイトが使用可能です.
A>TYPE DISKDATA.001↵ ……作成されたファイルの内容をTYPEコマンドで確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM     CONFIG   SYS     ATOK6A   SYS     ATOK6B   SYS     ATOK     DIC
PRINT    SYS     RSDRV    SYS     MOUSE    SYS     FORMAT   EXE     DISKCOPY EXE
CHKDSK   EXE     SYS      EXE     FC       EXE     FIND     EXE     MORE     COM
SORT     EXE     TREE     EXE     ATTRIB   EXE     EDLIN    EXE     README   DOC
DISKDATA 001
       21 個のファイルがあります.     このように「DIR/W↵」による表示出力が、そっくりディスク上の
   357376 バイトが使用可能です.       ファイルに格納されている
```

図9.5　リダイレクト「>」を使って、DIR コマンドの表示出力を内容とするディスク上のファイルを作成する

次は DIR コマンドの代わりに、CHKDSK コマンドの表示出力を内容とするディスクファイルを作成します。そのファイル名を「DISKDATA.002」としましょう(図9.6)。

```
A>CHKDSK > DISKDATA.002↵ ……同様に今回はCHKDSKコマンドによる表示出力を内容とする
                            ディスク上のファイル「DISKDATA.002」を作成する
A>DIR DISKDATA.002↵ ……作成されたファイルを確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

DISKDATA 002        231  89-07-30   13:28 ……確かに作成されている
        1 個のファイルがあります.
```

— 図9.6 —（次ページに続く）

```
A>TYPE DISKDATA.002        ……作成されたファイルの内容をTYPEコマンドで確認する

  1250304 バイト : 全ディスク容量
    95232 バイト : 2 個のシステムファイル
   797696 バイト : 22 個のユーザーファイル
   357376 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   454096 バイト : 使用可能メモリ
                         このようにCHKDSKコマンドの実行結果である、カレント
                         ドライブ(A:)のディスクの状態がファイルに格納されている
A>
```

図 9.6 リダイレクト「>」を使い、CHKDSK コマンドの表示出力を内容とするディスクファイルを作成する

次に、リダイレクト記号を2つ並べた「>>」(ダブルリダイレクト記号)を使い、コマンドの出力を、ディスク上にすでに存在するファイルに出力してみましょう。この場合は、既存ファイルの後ろに新データが追加され、**図 9.7** に示すような結果となります。CHKDSK コマンドの表示出力を、さきほど作成したファイル「DISKDATA.001」に出力してみましょう。

```
           ┌─注目
A>CHKDSK >> DISKDATA.001   ……図9.5で作成され、すでにディスク上に存在するファイル
                             「DISKDATA.001」に対して、「>>」記号を使って、CHKDSK
                             コマンドによる表示を出力する
A>DIR DISKDATA.001   ……実行後のファイルを確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

DISKDATA 001        659  89-07-30   13:29   ……図9.5と比較して作成日時が更新
                                              され、ファイルサイズが大きくなっている
        1 個のファイルがあります.
   356352 バイトが使用可能です.

A>TYPE DISKDATA.001   ……ファイル内容をTYPEコマンドで確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM   CONFIG  SYS   ATOK6A  SYS   ATOK6B  SYS   ATOK      DIC
PRINT    SYS   RSDRV   SYS   MOUSE   SYS   FORMAT  EXE   DISKCOPY  EXE
CHKDSK   EXE   SYS     EXE   FC      EXE   FIND    EXE   MORE      COM
SORT     EXE   TREE    EXE   ATTRIB  EXE   EDLIN   EXE   README    DOC
DISKDATA 001
       21 個のファイルがあります.
   357376 バイトが使用可能です.            こちらの内容は、図9.5で作成されたもの
```

— 図 9.7 — (次ページに続く)

```
1250304 バイト : 全ディスク容量
  95232 バイト : 2 個のシステムファイル
 798720 バイト : 22 個のユーザーファイル
 356352 バイト : 使用可能ディスク容量

 655360 バイト : 全メモリ
 454096 バイト : 使用可能メモリ

A>
```


図 9.7　ダブルリダイレクト記号「>>」を使って、すでに存在するディスクファイルに出力する

結果はこのように、すでに存在しているファイルの最後に、新しい内容が継ぎ足されています。同じことを繰り返せば、ファイルはどんどん継ぎ足されていくわけです。もし「>>」を使用したものの、指定したファイルがディスク上に存在していなかった場合は、指定したファイル名で新しいファイルが作成されます。

ところで今までは、

```
A>DIR ⏎
```

```
A>TYPE abcd.xyz ⏎
```

などのコマンドを実行する際に、リダイレクトのことはまったく意識しませんでしたが、実はこれらのコマンドは、暗黙のうちに

```
A>DIR >CON ⏎
```

```
A>TYPE abcd.xyz > CON ⏎
```

が実行されていたのです。つまり、出力先の指定がない場合には、自動的にCON(コンソール)が指定されるわけです。

さて、MS-DOSにおける、周辺装置間のデータ転送のまとめとして、同じ目的の操作を、異なる3つの方法で行ってみましょう。

例として、ドライブB：上の「README.DOC」という文字ファイルを、プリンタに出力して印字させてみます。つまり、ディスク上のファイルの内容を、デバイスファイルPRNに転送するだけの簡単なものです。これを、

- CTRL ＋ P によるプリンタへの出力機能
- COPY コマンドによるコピー機能
- リダイレクト「>」による、コマンド出力の転送機能

の3つの異なった手段で実行してみます。次の実行例のように、このいずれの方法によっても、同じような印字結果が得られます(図9.8、図9.9、図9.10)。

```
A>TYPE README.DOC
1.   バックアップ作成時の注意

  PS98－013(014)－HU／H4Wのシステムディスクは、9セクタ
フォーマット(720KB)でフォーマットされています。
                 ／H4Wのバックアップを作成する時には、
ご使用いただけます。

  なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品
に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
  使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけ
ます。

A>
```

ここで CTRL ＋ P を入力する

TYPEコマンドを CTRL ＋ P をONにして実行する

このようにディスプレイにもファイルの内容が表示される

```
1.   バックアップ作成時の注意

  PS98－013(014)－HU／H4Wのシステムディスクは、9セクタ
フォーマット(720KB)でフォーマットされています。
PS98－013(014)－HU／H4Wのバックアップを作成する時には、
                 ／9スイッチを指定してください。
  固定ディスクに対して、CHKDSK
ト: スキップセクタ」と表示される場合がありますが、
置がされていることを表しますので、「スキップセクタ」が表示されても安心して
ご使用いただけます。

  なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品
に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
  使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけ
ます。
```

画面への表示がこのように同時にプリンタにも出力される。その印字結果

図9.8　[CTRL]＋[P]を使ってディスク上のファイル内容をプリンタへ出力する

```
A>COPY README.DOC PRN ⏎ ……………COPYコマンドを実行して、ディスクファイルをPRN(プリンタ)へ転送する
        1 個のファイルをコピーしました. ……………転送終了のメッセージ

A>          ディスプレイには何も表示されない
```

↓

1.　バックアップ作成時の注意

　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタフォーマット（７２０ＫＢ）でフォーマットされています。
ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
…時に／９スイッチを指定してください。
…かされていることを表します…
ご使用いただけます。

　なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけます。

ファイル「README.DOC」がプリンタに転送された。その印字結果

図 9.9　COPY コマンドを使ってディスク上のファイル内容をプリンタへ出力する

```
A>TYPE README.DOC > PRN ⏎ ……………リダイレクトを使って、TYPEコマンドの出力をPRNへ送る
   ディスプレイには何も表示されない
A>
```

↓

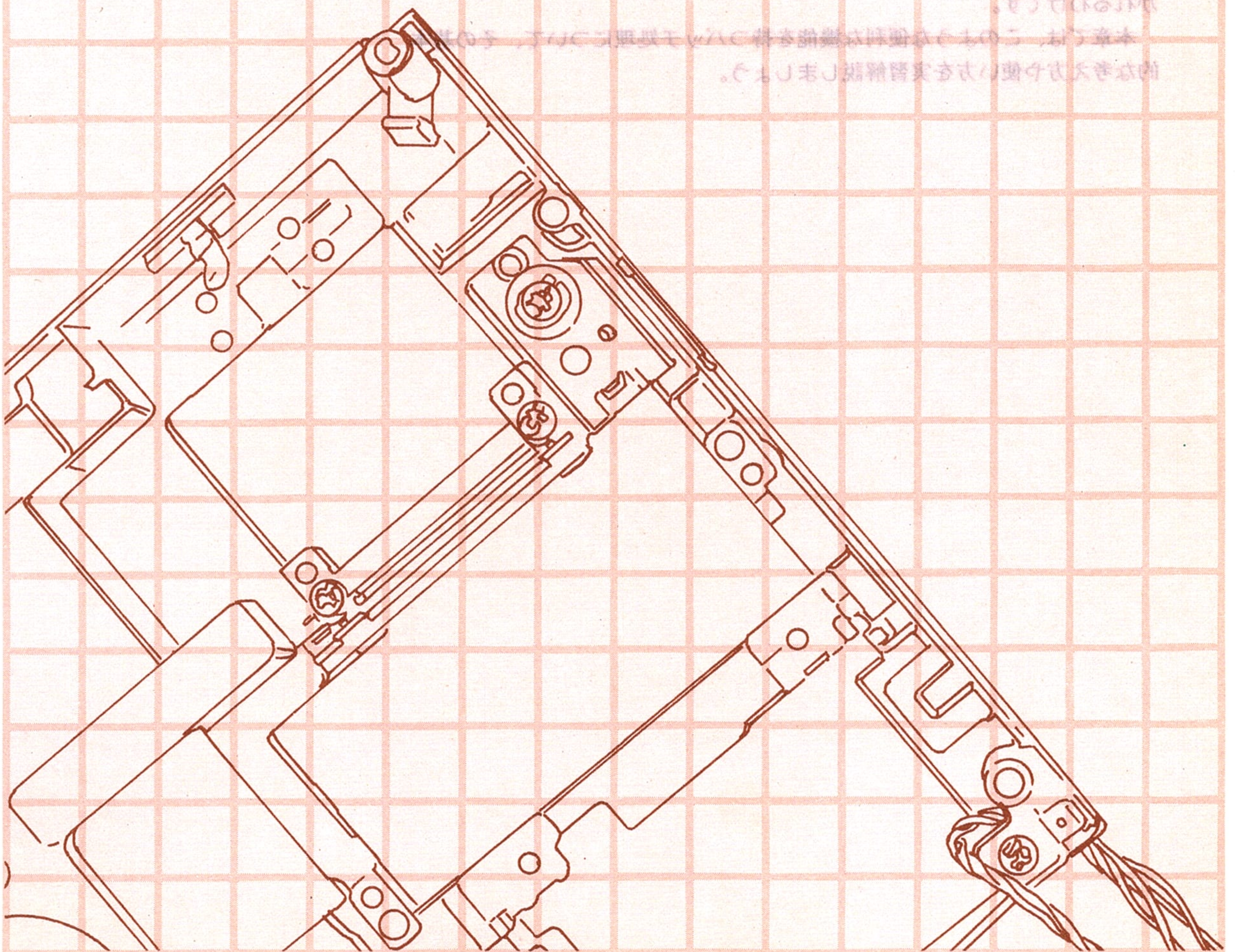
1.　バックアップ作成時の注意

　ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのシステムディスクは、９セクタフォーマット（７２０ＫＢ）でフォーマットされています。
ＰＳ９８－０１３（０１４）－ＨＵ／Ｈ４Ｗのバックアップを作成する時には、
…時に／９スイッチを指定してください。
…かされていることを表します…
ご使用いただけます。

　なお、「スキップセクタ」が表示された場合、使用可能ディスク容量を、本製品に添付されている「固定ディスクを使用する場合のご注意」で確認してください。
　使用可能ディスク容量が記載されている値以上であれば、正常にご使用いただけます。

TYPEコマンドの出力がプリンタに転送された。その印字結果

図 9.10　TYPE コマンドとリダイレクトを使ってディスク上のファイル内容をプリンタへ出力する

10章
複数のコマンドやプログラムの自動実行(バッチ処理)

MS-DOS には、バッチ処理という便利な機能があります。たとえば、私たちが毎日の仕事に使うプログラムを、いつも同じ手順で立ち上げる場合や、一連のコマンドやプログラムを、同じような手順で何度も繰り返し実行しなければならない場合など、このバッチ処理の機能をうまく利用することにより、たいへん能率的に、かつ楽に仕事を行うことができます。

これは、複数のコマンドやプログラムから成る一連の作業の手順を書き並べたファイルを用意しておき、実際に作業を行うときには、そのファイル名をキー入力するだけで、そこに書かれている内容を自動的に実行させる機能です。また、バッチ処理の特別な機能である「自動スタート機能」を使えば、ディスクをセットして、リセットボタンを押すだけで、あとはキーボードに触れることなく自動的に目的のプログラムを立ち上げることができます。つまり、オフィスで使うビジネスソフトなどが、リセットボタンを押すだけの操作で自動的に立ち上がり、そのまますぐ仕事に取りかかれるわけです。

本章では、このような便利な機能を持つバッチ処理について、その基本的な考え方や使い方を実習解説しましょう。

10.1　バッチ処理の概念

バッチ処理とは、ユーザーがあらかじめ作成しておく「実行指示書」に書かれた、1つ、あるいは複数のコマンドやプログラムなどから成る一連の操作を、MS-DOSが自動的に実行する機能です。つまり、あらかじめ指示しておいた複数の仕事を、MS-DOSが「一括処理」してくれるわけです。その実行指示書にあたるのが、**バッチファイル**と呼ばれるファイル(文字ファイル)であり、それぞれのユーザーが、エディタプログラムやCOPYコマンド、あるいはワープロソフトなどを使って任意に作成します。

バッチファイルのファイル名には、自由に名前を付けられますが、そのファイルタイプは必ず、

.BAT

でなければなりません。たとえばABCDというファイル名であれば、

ABCD.BAT

となるわけです。

こうして作られたバッチファイルは、実行する場合、通常のプログラムファイル(マシン語ファイル)と同じような扱いを受けます。もちろんこのファイルの内容は、コマンド名やプログラム名などを並べただけの文字ファイル(テキストファイル)ですが、「.COM」や「.EXE」のファイルタイプが付いた、

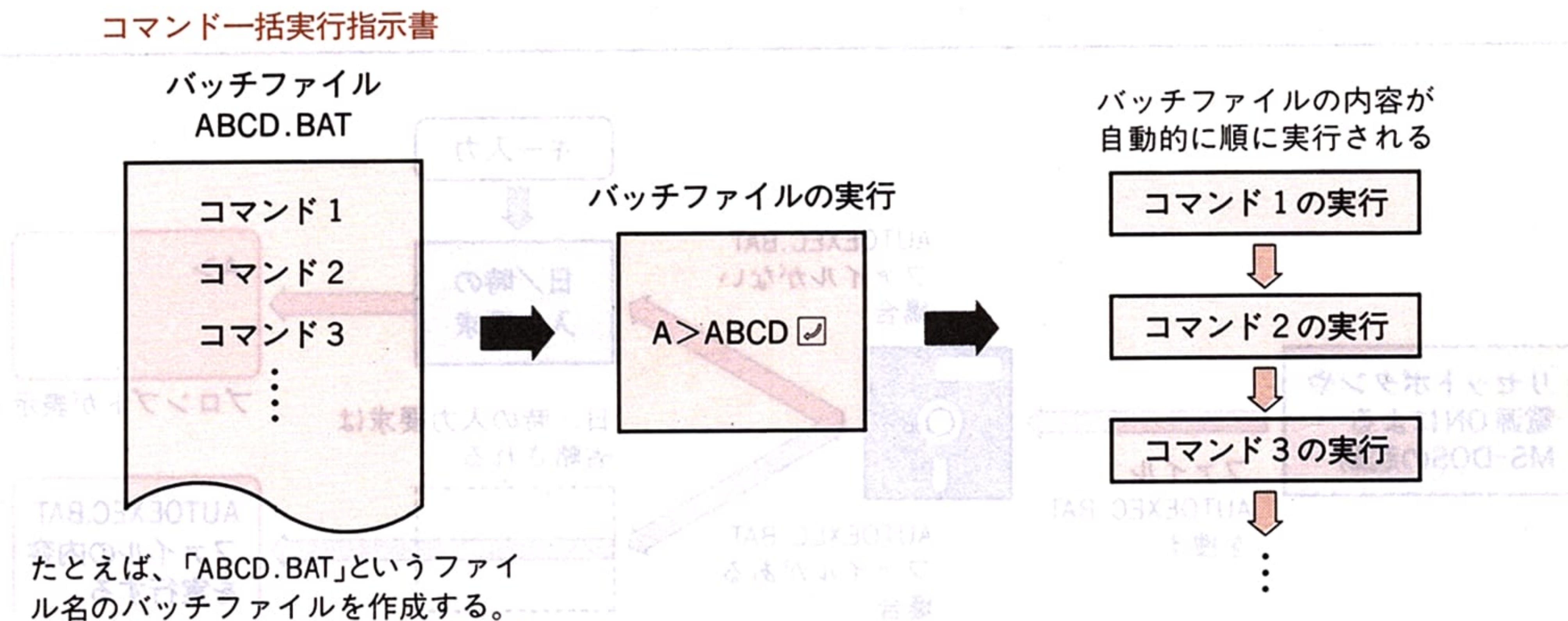



図 10.1　バッチ処理における実行の流れ

実行可能なマシン語ファイルと同じように実行することが可能です。つまり、バッチファイル「ABCD.BAT」の内容に記された一連のコマンドやプログラムなどを、

```
A>ABCD ⏎
```

のように、その主ファイル名をキー入力し、⏎することにより、自動的に実行させることができるのです(図 10.1)。

■ 自動スタートのためのバッチファイル

バッチファイルの主ファイル名は、どのようなものであろうと(もちろん一般的なファイル名の制限の範囲で)任意ですが、

```
AUTOEXEC.BAT
```

というファイル名には特別の意味があります。

この名前のバッチファイル —— **自動スタートバッチファイル**(オートスタートバッチファイル)がディスク上に存在すると、MS-DOS は起動時にそのファイルの内容を自動的に実行します。つまり、MS-DOS が起動した直後、MS-DOS はディスク上の「AUTOEXEC.BAT」という名前のファイルを捜し、もしそれが存在すれば、そのバッチファイルの内容を自動的に実行します。その場合は、MS-DOS の通常の起動時に必ず行われる日／時の問い合わせが省略されますので、ユーザーはディスクをセットしてリセットボタンを押すだけで、AUTOEXEC.BAT ファイルに書かれた目的のプログラムなどを自動的に(手放しで)立ち上げることができるわけです。またこの自動スタートバッチファイルの機能は、単にコマンドやプログラムを実行するだけでなく、コンピュータシステムの各種の環境の初期設定などに利用することも広く一般に行われています(図 10.2)。

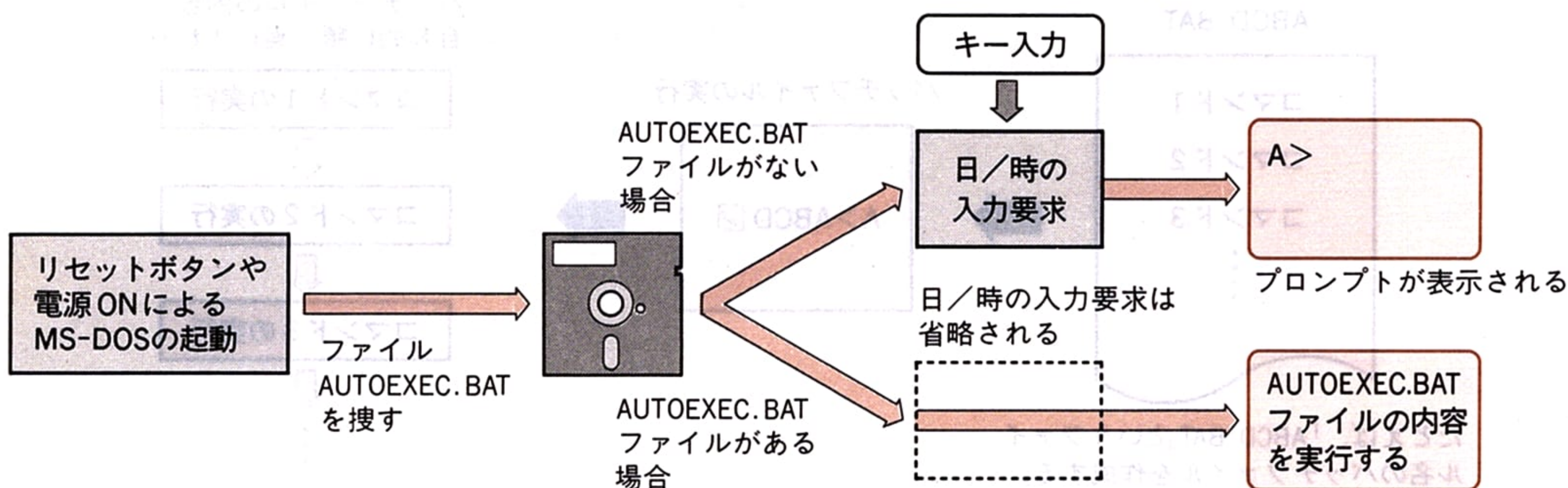



図 10.2　AUTOEXEC.BAT ファイルによる自動スタートの実行の流れ

また、この AUTOEXEC.BAT ファイルも、バッチファイルの１つであることに変わりありませんので、通常のバッチファイルと同様に、MS-DOS のコマンドレベルにおいて、「AUTOEXEC ⏎」とキー入力することによっても、その内容を実行することができます。

では、自動スタートバッチファイルの作成を次節で、また通常のバッチファイルの作成をその次の節で実習しましょう。

10.2　MS-DOS の起動時に自動実行される バッチファイルの作成

前節で述べたように、MS-DOS は起動時に必ず、ディスク上の「AUTOEXEC.BAT」というファイルを捜します。そして、もし存在すれば、そのファイルの内容が自動的に実行されます。つまり、AUTOEXEC.BAT（AUTO：自動、EXEcute：実行する、BATch：バッチファイル）というファイルを利用すれば、任意のコマンドやプログラムなどを（内蔵コマンドでも、ビジネスプログラムなどの外部プログラム（コマンド）でも）、ディスクをセットしてリセットボタンを押すだけで自動的に実行することができるわけです。

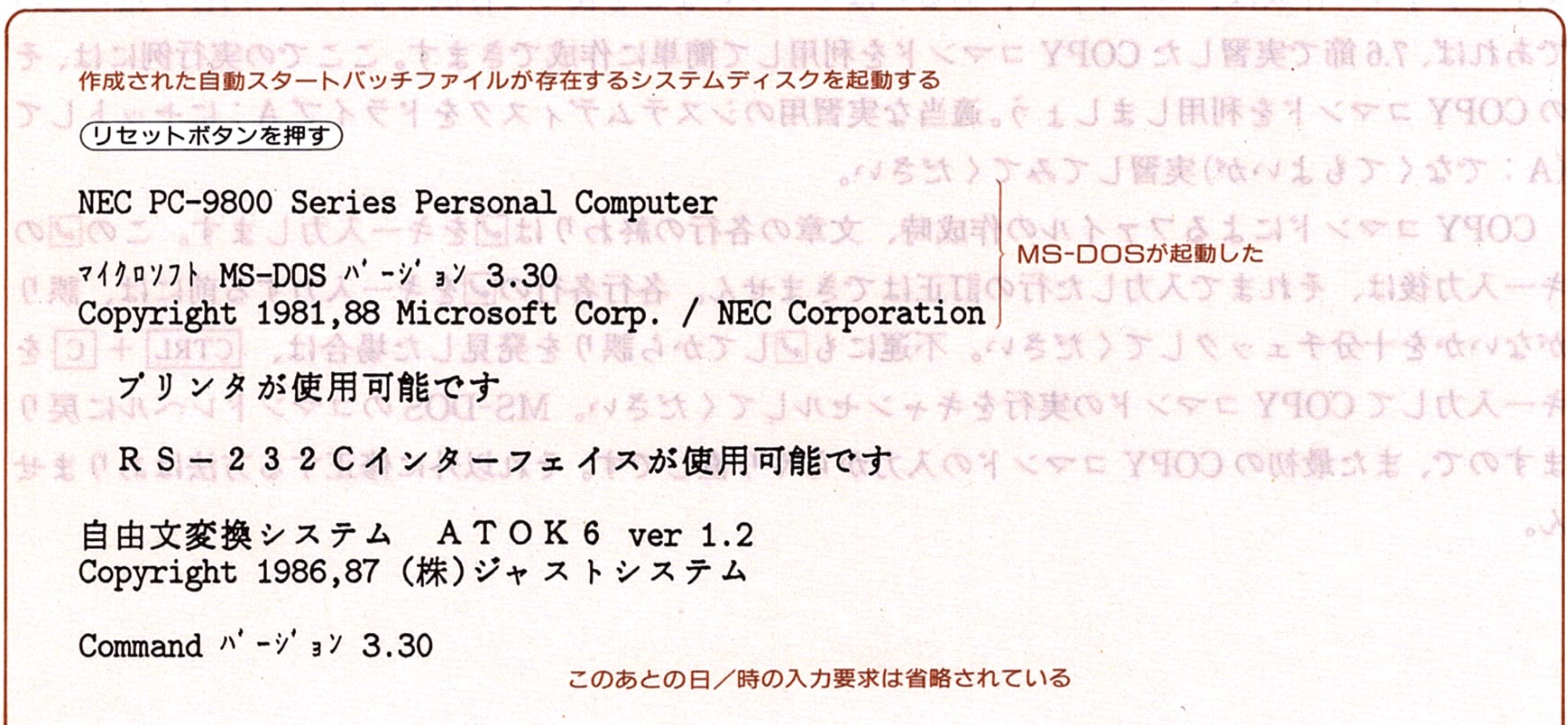
この**自動スタート**の機能を利用するには、AUTOEXEC.BAT というファイル名で、実行したいコマンドやプログラムなどの、コマンドラインの記述を書き並べたファイルを作成しなければなりません。文字ファイルの作成は、エディタや、あるいはワープロなどを使って作成しますが、内容が短いものであれば、7.6 節で実習した COPY コマンドを利用して簡単に作成できます。ここでの実行例には、その COPY コマンドを利用しましょう。適当な実習用のシステムディスクをドライブ A：にセットして（A：でなくてもよいが）実習してみてください。

COPY コマンドによるファイルの作成時、文章の各行の終わりは⏎をキー入力します。この⏎のキー入力後は、それまで入力した行の訂正はできません。各行各行の⏎をキー入力する前には、誤りがないかを十分チェックしてください。不運にも⏎してから誤りを発見した場合は、CTRL ＋ C をキー入力して COPY コマンドの実行をキャンセルしてください。MS-DOS のコマンドレベルに戻りますので、また最初の COPY コマンドの入力からやり直しです。それ以外に修正する方法はありません。

ここで作成する AUTOEXEC.BAT ファイルの内容は、MS-DOS が起動すると同時に、カレントドライブ(つまり MS-DOS を起動したディスク)に対して DIR コマンドを実行し、そのあとにコマンドレベルになるという非常に簡単なものです(図 10.3、図 10.4)。

```
A>COPY CON AUTOEXEC.BAT ⏎ ……COPYコマンドにより、コンソール入力(キー入力)を「AUTOEXEC.BAT」
                             というファイルにコピーする。つまりファイルを作成する
DIR/W ⏎ ……作成するファイル内容を入力する
^Z ⏎ ……CTRL+Z でCOPYコマンドを終了する。「^」+「Z」ではなく CTRL キーを押しながら Z をキー入力する
        1 個のファイルをコピーしました. ……ファイルが作成された

A>DIR AUTOEXEC.BAT ⏎ ……作成されたファイル「AUTOEXEC.BAT」を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

AUTOEXEC BAT         7  89-07-30   13:33 ……作成されている
        1 個のファイルがあります.
   357376 バイトが使用可能です.

A>TYPE AUTOEXEC.BAT ⏎ ……TYPEコマンドでその内容を画面に表示して確認する
DIR/W ……バッチファイルの内容

A>
(ここの「DIR/W」を「MIFES」に変更して、図4.10のディスク上に作成すれば、自動スタートのMifes-98システムディスクができる)
```

図 10.3　COPY コマンドを利用して、自動スタートバッチファイルを作成する

```
作成された自動スタートバッチファイルが存在するシステムディスクを起動する
(リセットボタンを押す)

NEC PC-9800 Series Personal Computer                     ┐
                                                         │ MS-DOSが起動した
ﾏｲｸﾛｿﾌﾄ MS-DOS ﾊﾞｰｼﾞｮﾝ 3.30                              │
Copyright 1981,88 Microsoft Corp. / NEC Corporation      ┘

  プリンタが使用可能です

  ＲＳ－２３２Ｃインターフェイスが使用可能です

自由文変換システム　ＡＴＯＫ６ ver 1.2
Copyright 1986,87 (株)ジャストシステム

Command ﾊﾞｰｼﾞｮﾝ 3.30
                    このあとの日／時の入力要求は省略されている
```

— 図 10.4 — (次ページに続く)

```
A>DIR/W ……………自動スタートバッチファイルの内容であるDIRコマンドの実行が自動的に行われる

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM     CONFIG   SYS     ATOK6A   SYS     ATOK6B   SYS     ATOK     DIC
PRINT    SYS     RSDRV    SYS     MOUSE    SYS     FORMAT   EXE     DISKCOPY EXE
CHKDSK   EXE     SYS      EXE     FC       EXE     FIND     EXE     MORE     COM
SORT     EXE     TREE     EXE     ATTRIB   EXE     EDLIN    EXE     README   DOC
AUTOEXEC BAT ……………これが作成した自動スタートバッチファイル
        21 個のファイルがあります.
    357376 バイトが使用可能です.

A>
A> ……………DIRコマンドが実行されてからMS-DOSのプロンプトが表示され、コマンドレベルになった
```

図 10.4　作成された自動スタートバッチファイルを実行する

ここでのバッチファイルの内容は、DIR コマンドを実行するだけのものですが、一連のコマンドやプログラムなどを実行させるには、次節の例のように、複数のコマンドやプログラムを書き連らねればよいわけです。例として、ワープロソフト一太郎の、オリジナルのシステムディスクに含まれている自動スタートバッチファイルの内容を TYPE コマンドで表示してみましょう(図 10.5)。

```
A>TYPE AUTOEXEC.BAT ⏎ ……………一太郎の自動スタートバッチファイルの
                               内容をディスプレイに表示する
ECHO OFF ……………(後述)
DATE ……………DATEコマンドを実行する
TIME ……………TIMEコマンドを実行する
ECHO ON ……………(後述)
JXW ……………一太郎JXW.COMを実行する
A>
```
(ECHO OFF ～ JXW：バッチファイルの内容)

図 10.5　ワープロソフト一太郎の自動スタートバッチファイルの内容

この一太郎のシステムディスクを立ち上げる際には、AUTOEXEC.BAT ファイルが存在しているにもかかわらず、日／時の入力を要求してきますが、それは、その自動スタートバッチファイルの内容に書かれている DATE コマンドと TIME コマンドが実行されるからです(AUTOEXEC.BAT ファイルが存在する場合は、MS-DOS 起動時の自動的な日／時の入力要求は省略される)。もし、リセットボタンを押すだけで、あとは手放しで一太郎を立ち上げたいのであれば、この 2 つのコマンドを削除した自動スタートバッチファイルを作ればよいわけです。

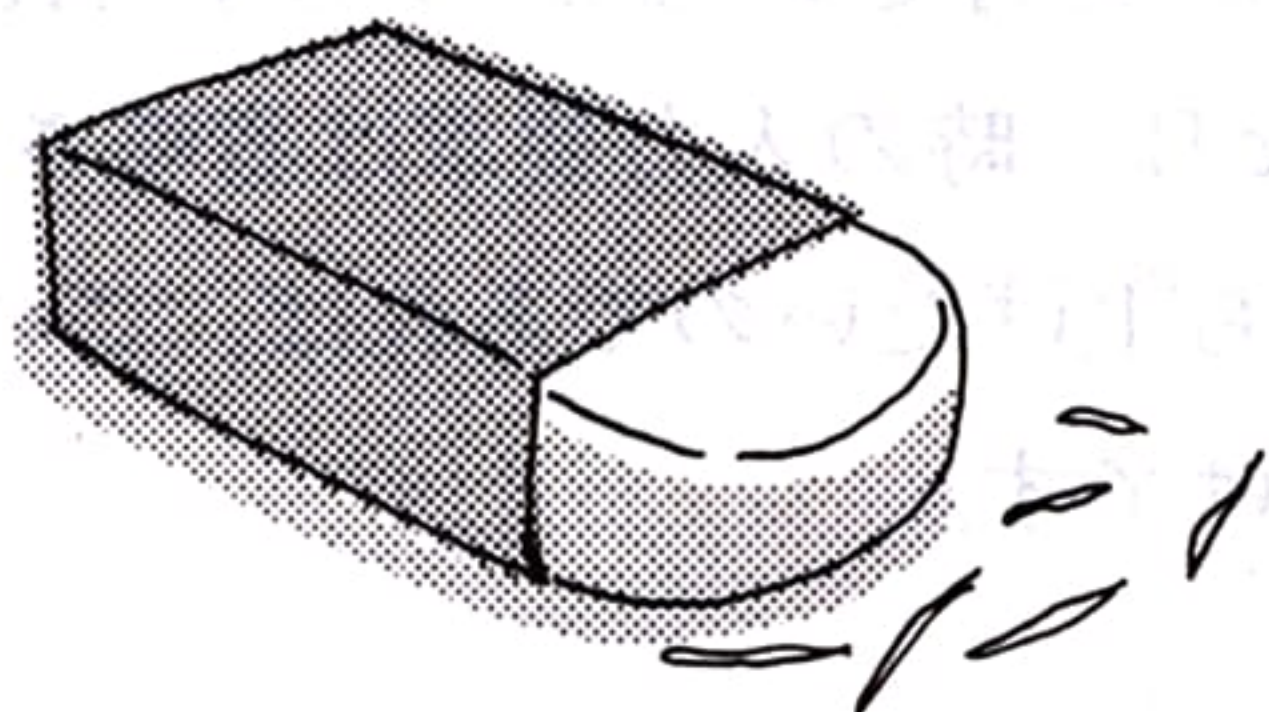
また、この例の中の「ECHO」というコマンド(本書では実習していない)は、このように使った場合、最初の「ECHO　OFF」と次の「ECHO　ON」の間にある、DATEコマンドやTIMEコマンドのコマンドライン(「A>DATE」や「A>TIME」)が画面に表示されないようにするためです。しかし、そのコマンドラインが表示されても一向に差しつかえありませんので、このバッチファイルに「ECHO OFF」や「ECHO ON(または省略して「ECHO」)」などを書く必要はまったくありません。また、「DATE」や「TIME」コマンドなども、日時を毎回修正する必要があるわけでなく、これなども書いておく必要はまったくありません。結局、このバッチファイルには、「JXW」の1行だけで十分なのです(ビジネスソフトなどに付属の自動スタートバッチファイルには、なぜかこのような不用なコマンドが記述されている例が多い)。

10.3　通常のバッチファイルの作成

ここでは、簡単な内容のバッチファイルを作成して、その一連の内容を実行させてみましょう。内容は次のようなものにします。

① DIRコマンドを実行して、ファイルタイプが「.BAT」であるファイルを表示する
② TYPEコマンドで、その(自分自身の)バッチファイル「TS_BATCH.BAT」の内容を画面に表示する
③ PROMPTコマンドで、プロンプト「A>」「B>」などに、日／時が表示されるように設定する

このバッチファイルを実行すると、①→②→③と、それぞれの内容が自動的に実行され、全部の実行が終了すると、通常のコマンドレベルに戻ります。

では、このバッチファイルの名前を「TS_BATCH.BAT」として、ファイルを作成しましょう(図10.6)。

「コンソールのキー入力」を「ファイル」に「コピーせよ」

作成するバッチファイルの内容

```
A>COPY CON TS_BATCH.BAT ⏎
DIR *.BAT ⏎ ………①ファイルタイプが「.BAT」であるすべてのファイルを表示する
TYPE TS_BATCH.BAT ⏎ ………②ファイル「TS_BATCH.BAT」の内容を表示する
PROMPT $_$D$_$T$_$N$G ⏎ ………③プロンプトに日/時の表示を設定する
^Z ⏎ ………CTRL+Z を入力してCOPYコマンドを終了させる

A>
```

図 10.6　COPY コマンドを利用して、バッチファイル「TS_BATCH.BAT」を作成する

これでバッチファイル、TS_BATCH.BAT が作成されました。DIR コマンドで確認してみましょう(図 10.7)。

```
A>DIR TS_BATCH.BAT ⏎ ………作成されたファイルを確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

TS_BATCH BAT        53  89-07-30   13:38 ………ファイル「TS_BATCH.BAT」が作成されている
         1 個のファイルがあります.
    356352 バイトが使用可能です.

A>
```

図 10.7　作成されたバッチファイルの確認

ディスク上にファイル「TS_BATCH.BAT」が作成されていることが確認できました。その内容は、TYPE コマンドを使って見ることができますが、このバッチファイルの実行の中で行われますので、ここでは誤りはないものとして、いきなりこのバッチファイルを実行してみましょう。実行は、通常のマシン語のプログラムのように、その主ファイル名をキー入力します(図 10.8)。

```
A>TS_BATCH ↵

A>DIR *.BAT

 ドライブ A: のディスクのボリュームラベルは DOS_WORK
 ディレクトリは A:¥

TS_BATCH BAT        53  89-07-30   13:38
AUTOEXEC BAT         7  89-07-30   13:33
        2 個のファイルがあります.
   364544 バイトが使用可能です.

A>TYPE TS_BATCH.BAT
DIR *.BAT
TYPE TS_BATCH.BAT
PROMPT $_$D$_$T$_$N$G

A>PROMPT $_$D$_$T$_$N$G


1989-07-30 (日)
13:40:15.00
A>↵

1989-07-30 (日)
13:40:15.00
A>
```

A>TS_BATCH ……バッチファイル「TS_BATCH.BAT」を実行する

A>DIR *.BAT ……バッチファイルに登録されているDIRコマンドが自動的に実行される

A>TYPE TS_BATCH.BAT ……同じくTYPEコマンドが実行される

DIR *.BAT / TYPE TS_BATCH.BAT / PROMPT $_$D$_$T$_$N$G：現在実行されている自分自身のバッチファイル「TS_BATCH.BAT」の内容が表示された

A>PROMPT $_$D$_$T$_$N$G ……同じくPROMPTコマンドが実行される

1989-07-30 (日) / 13:40:15.00 / A>：これ以降、プロンプトはこのような表示になる

①の部分が実行された

②の部分が実行された

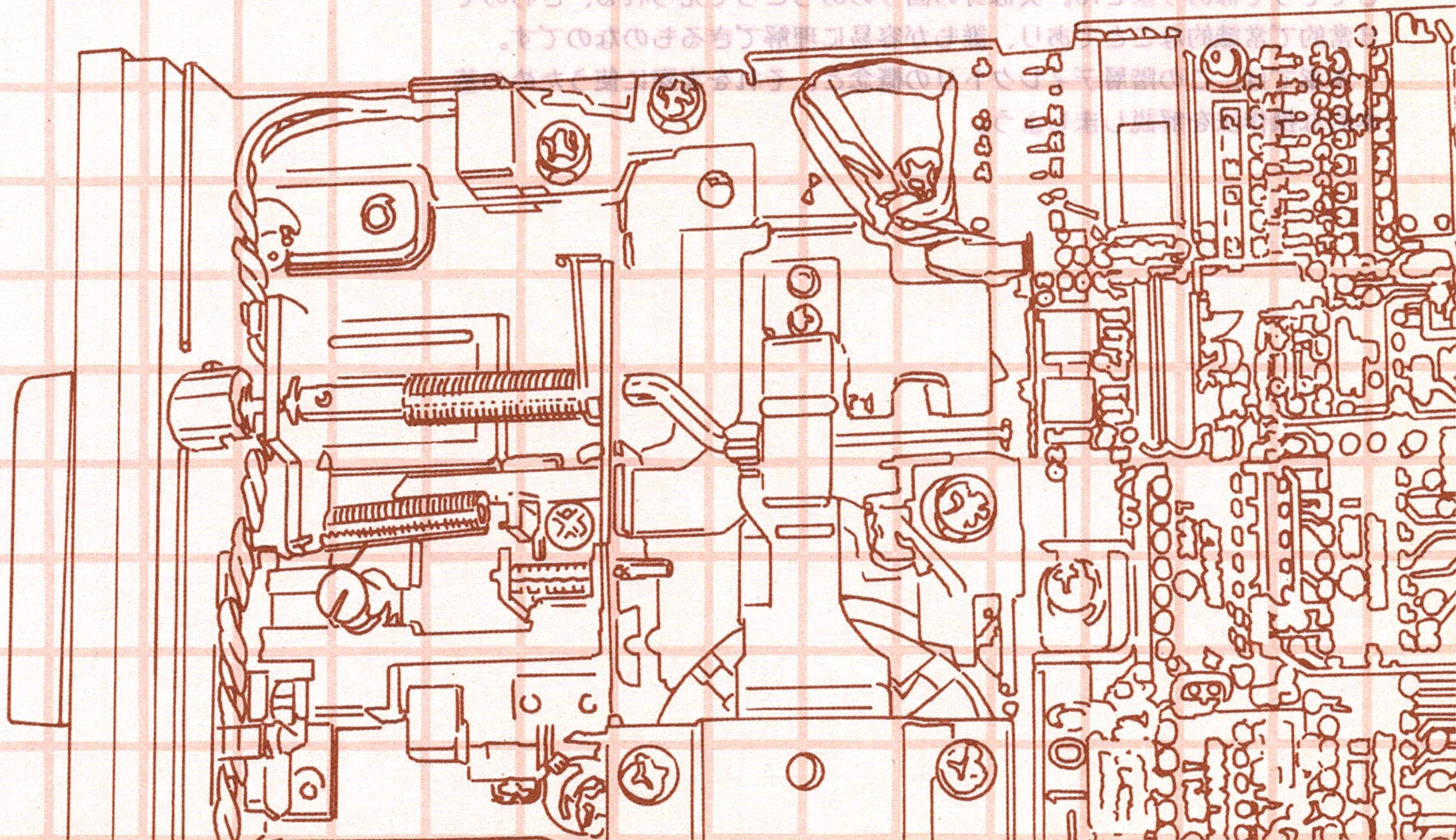
③の部分が実行された

バッチファイルに登録されている各種コマンドが自動的に実行されていく

図 10.8　バッチファイル「TS_BATCH.BAT」を実行する

このように、一連のコマンドやプログラムが自動的に実行されました。

ここでの実習はごく基本的なものですが、バッチ処理の機能には、本書では触れていないさらに多くの機能があります。たとえば、メッセージの出力、条件付きの実行(条件分岐が可能)、バッチファイル内での仮ファイル名を実行時に実ファイル名で実行する機能など、これらを組み合わせたバッチプログラム(バッチファイルのことであるが、複雑なものは、一見BASICのプログラムのようである)を作成することにより、たいへんユーザーフレンドリーな対話形式の問い合わせ画面が実現できるなど、このバッチ機能をさまざまな用途に応用することができます。

11章
ファイルの組織的な分類整理法

本章はたいへん重要です。

本書では今まで、ディスクを１つの容器と考え、その中に各種のファイルを収容してきました。つまり、１つの容器の中に、いろいろなファイルをゴチャゴチャに詰め込んでいたわけです。

このような使い方でも、記憶容量の小さなフロッピーディスクや、収容するファイルの数が少ない場合は、あまり混乱は起きません。なぜなら、フロッピーディスク程度の小さな容量では、必然的にソフトやファイルの種類を分けて、ディスク単位で分類整理をしているからです。しかし、ディスクの容量が大きくなったり、多くのファイルを１つのディスクに収容するような場合は、何らかの方法でファイルを分類整理して収容しないことには、いろいろな面で支障が出てきます。

たとえば、目的のファイルを捜し出すのに苦労する、当面の仕事に関係ないファイルが同居しているので、作業がやりにくい、関係のないファイルを誤って書き換えたり、消したりする危険性がある、まったく別の仕事であっても同じ名前のファイルが作れないなど、多くの問題点が考えられるでしょう。そこでMS-DOSをはじめ、実務向けのOSのほとんどは、「階層ディレクトリ」あるいは「ツリー(木)構造ディレクトリ」と呼ばれるファイルの管理法を採用しています。ファイルを階層的に管理することの必要性は、とくにハードディスクでは決定的です。

階層構造の概念は、一見、難しそうな印象を与えているようですが、けっしてそうではありません。実は身の回りのあちこちで見られる、きわめて日常的で常識的なことであり、誰もが容易に理解できるものなのです。

本章では、この階層ディレクトリの概念と、それを実際に使うための基本的な操作法を解説しましょう。

11.1　階層ディレクトリの概念

まず、**図 11.1** をご覧ください。これは、大きな容器の中に小さい容器があり、またその中にも容器があり、丸い豆がそれらの容器にいくつかはいっている状態を表しています。

この絵を次のように考えてください。

- この図のいちばん大きな容器を、1 枚のディスク全体と考える。これは大もとの容器、**ルートディレクトリ**と呼ばれる
- その中に含まれる大小いろいろな容器は、すべて**ディレクトリ**であり、**子ディレクトリ**とか**サブディレクトリ**とか呼ばれる
- 丸い豆は各種の**ファイル**を表す

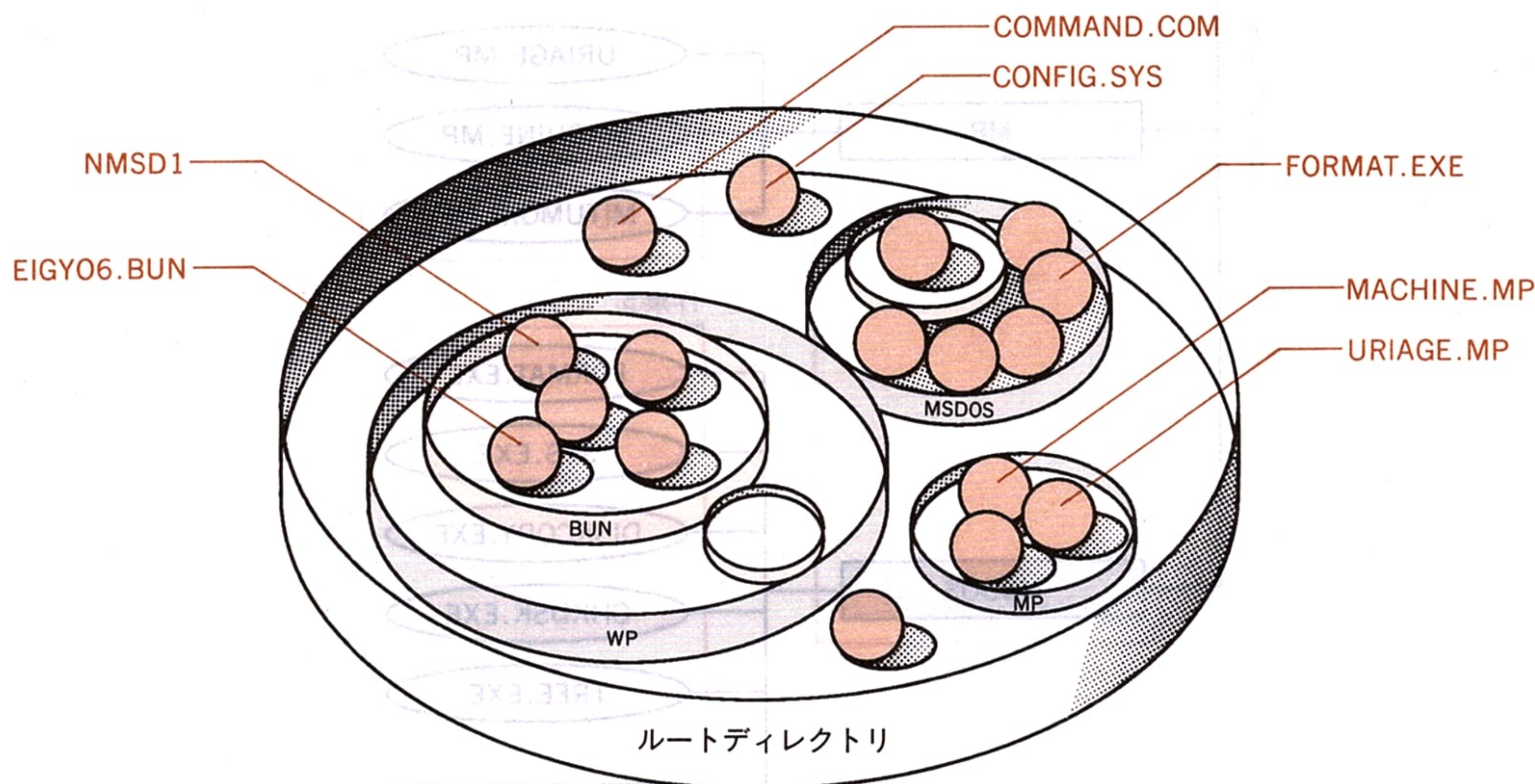



図 11.1　階層ディレクトリの考え方を示す模式図

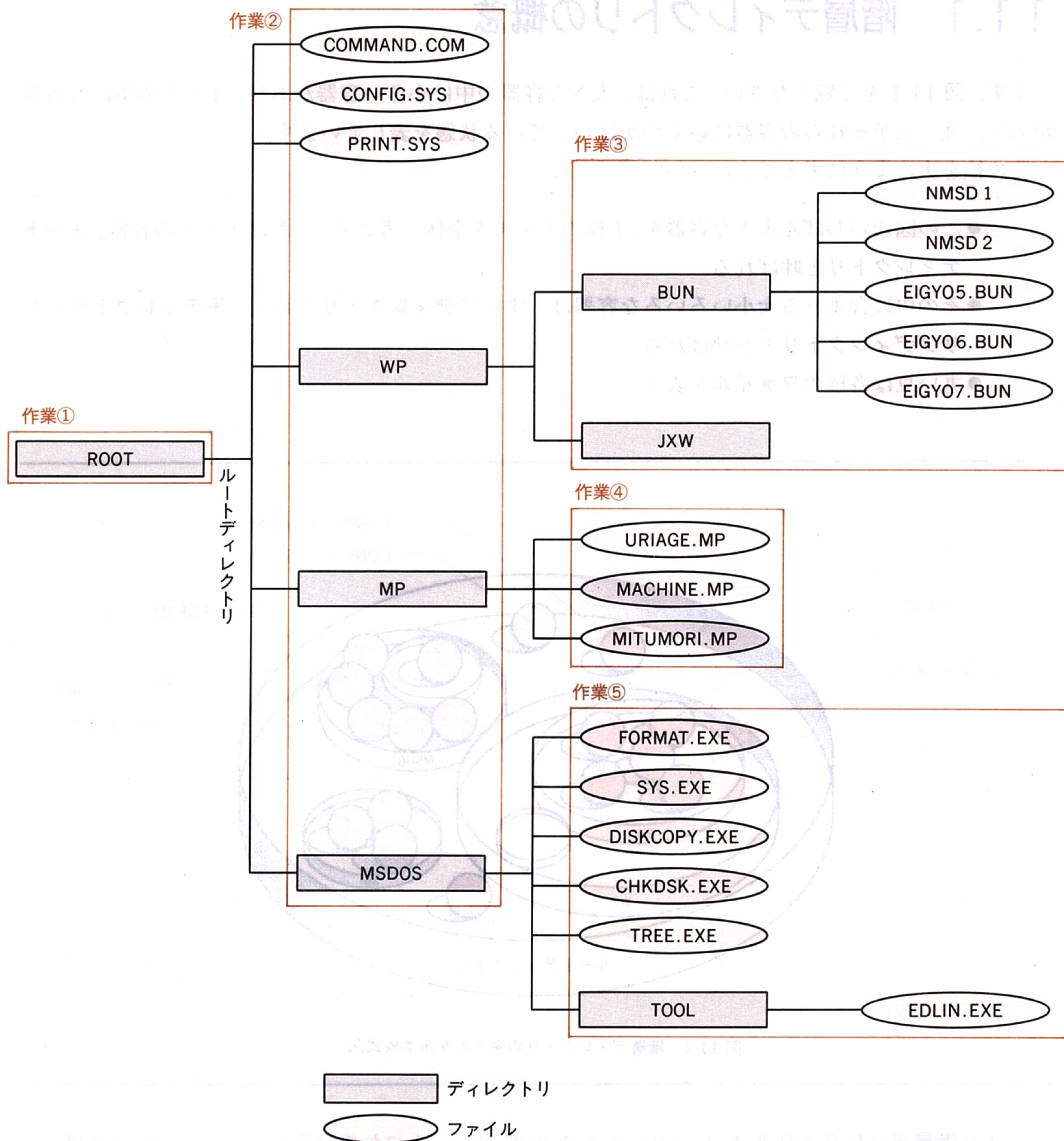
つまり**階層ディレクトリ**とは、1 つのディスク(容器)を、いくつかの容器(ディレクトリ)に階層的に(親→子→孫→…、あるいは、部→課→係→…、のように)分け、各種のファイルを分類して、それぞれのディレクトリ(容器)に整理して収容するという考え方です。つまり、多くのファイルの中から、関連するファイルだけを集めて 1 つの容器(ディレクトリ)に格納するわけです。

作業②
COMMAND.COM
CONFIG.SYS
PRINT.SYS
WP
MP
MSDOS

作業①
ROOT
ルートディレクトリ

作業③
BUN
NMSD 1
NMSD 2
EIGYO5.BUN
EIGYO6.BUN
EIGYO7.BUN
JXW

作業④
URIAGE.MP
MACHINE.MP
MITUMORI.MP

作業⑤
FORMAT.EXE
SYS.EXE
DISKCOPY.EXE
CHKDSK.EXE
TREE.EXE
TOOL
EDLIN.EXE

ディレクトリ
ファイル

※この図は、ルートディレクトリを左にして、右に向かって成長する木として表現している

図 11.2　実際のファイルが収容されている階層ディレクトリと実習の作業区分

図 11.1 のように書くと、階層が深くなる（重ねる）に従って、容器がだんだん小さくなりますが、ディスク上では、そのようなことはありません。どの容器も大きさの差別や制限はありません。大もとの容器以外は、大きなファイルをたくさん収容すればするだけ、どんどん大きく膨らみます（風船のように）。それらの「容器」の中で大もとの容器は、ディスク上のすべてのものを収容するディレクトリ――つまり最初の階層のディレクトリであり、**ルートディレクトリ**と呼ばれます。

次に、実際のいろいろなファイルが、階層的に整理されて各ディレクトリに収容されているディスクの例を示しましょう。**図 11.2** は、ルートディレクトリを左にして、右に向かって成長する「木」のように描かれています。階層構造を**ツリー構造**（tree：木）とも呼ぶのは、まずルートディレクトリ（Root：根）があって、そこから、幹→枝→小枝→…と、木が成長していくような構造になるからです。幹、枝、小枝などは「子ディレクトリ」に当たり、そのそれぞれに生えている「葉」が「ファイル」であると考えればよいでしょう。（なお、図 11.2 には、のちほど実習する作業の作業区分も示してある）

図 11.2 のディレクトリ構造を簡単に説明しておきましょう。

まず最初の階層であるルートディレクトリ（大もとの容器）に注目してみましょう。

ルートディレクトリには、3 個のファイルと、3 つのディレクトリ（子供の容器に当たるもの）が収容されています。このディスクのルートディレクトリを DIR コマンドで見ると**図 11.3** のようになります。

```
A>DIR ↵  ……………図11.2のディスクのルートディレクトリの内容を表示する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥
                                          ─ファイル
COMMAND  COM     24931  88-07-13   0:00 ……コマンド・プロセッサ
CONFIG   SYS        87  89-07-29  17:25 ……システム構築用MS-DOSのファイル
PRINT    SYS      5855  88-07-13   0:00 ……プリンタを動作させるためのデバイスドライバ（MS-DOSバージョン.2xでは不要）
WP           <DIR>      89-07-30  15:21 ……ワープロの文書ファイルのためのディレクトリ
MP           <DIR>      89-07-30  15:21 ……Multiplanのデータファイルのためのディレクトリ
MSDOS        <DIR>      89-07-30  15:21 ……日常よく使うMS-DOSコマンドのためのディレクトリ
        6 個のファイルがあります.
   747520 バイトが使用可能です.            ─ディレクトリ

A>
```

図 11.3　図 11.2 の階層ディレクトリを持つディスクのルートディレクトリの内容を DIR コマンドで見る

これがルートディレクトリ（最初の階層）の内容です。このように、ルートディレクトリ上のファイルは、MS-DOS のシステム関係の 3 つだけしか存在していません（このディスクは MS-DOS のシステムディスクである）。

一般に、階層ディレクトリを利用する場合、MS-DOSのシステムディスクであれば、『ルートディレクトリには、MS-DOSシステム関係のファイル以外は置かない』ことが理想的です。日本語入力システムもルートディレクトリに置かず、可能であればFEP(フロントエンド・プロセッサ：日本語入力システムのこと)用のディレクトリを作成して、そこに置く方がよいでしょう。つまり、ルートディレクトリには「COMMAND.COM」と「CONFIG.SYS」と「AUTOEXEC.BAT」以外のファイルは置かないことが理想なのです。したがって、MS-DOSのシステムディスクでないデータディスクであれば、ルートディレクトリには何も置かないことになります。

重要！　「ファイル」と、それを収容する容器である［ディレクトリ］とを明確に区別して考えてください。

次に、ルートディレクトリに収容されている3つのディレクトリを順に見ていきましょう。

まず、[WP]というディレクトリに注目してみましょう。このディレクトリは、ワープロの文書ファイルを収容するために使います。このディレクトリには、ファイルは収容されておらず、さらに2つの子ディレクトリが作られています。新松の文書ファイルを収容する［BUN］と、一太郎の文書ファイルを収容するための［JXW］の2つです。この［BUN］と［JXW］は、[WP］を「親ディレクトリ」とする「子ディレクトリ」に当たります。それぞれの文書ファイルは、これらのディレクトリに収容されています。

次に、[MP］というディレクトリに注目してみましょう。このディレクトリは、表計算ソフトのMultiplanのデータファイルを収容するために使います。ここには、子ディレクトリは作られておらず、いくつかのデータファイルが直接収容されています。

次に、[MSDOS]というディレクトリに注目してみましょう。このディレクトリは、MS-DOSの外部コマンドの中の、日常よく使うプログラムファイルを収容しておくために使います。またこのディレクトリには、各種の「道具」(各種のユーティリティプログラムなど)を入れておくためのディレクトリ［TOOL］が作られています。

さて、今後はこのような階層構造を持つディスクを例に、解説を進めていきましょう。次節では、ここで示した階層ディレクトリのディスクを実際に作成してみることによって、その作成過程や、階層ディレクトリにおけるファイルの運用法などを実習します。当然ながら階層ディレクトリは、それぞれのディスクごと、フロッピーディスクの1枚ごとに独立したものであり、ほかのディスクにはまったく関係なく、独自の構造を持つことができます。

11.2　階層ディレクトリの作成法

では、図11.2に示した階層ディレクトリとそっくり同じものを実際に作成しながら、その作業に関連するコマンドや機能を解説しましょう。ここでの実習から、階層ディレクトリの概念とその運用法の概略を学んでください(運用法については、次節で改めて解説する)。全体の作業は一見複雑そうに見えますが、コマンドの１つひとつは単純なものばかりですので安心してください。

11.2.1　階層ディレクトリの作成実習

ここでの実習は、先の図11.2に示したように、全体の作業を、階層別およびディレクトリ別に区分して、それぞれのブロックごとに行います。

では、区分の順に作業を始めましょう。

【作業1】実習用ディスクの下準備(ルートディレクトリの準備)

では、ドライブA：にMS-DOSシステムディスク、ドライブB：には内容を消去してもよい実習用の適当なディスクをセットして、/Sスイッチ付きFORMATコマンドを実行するなどの方法で、ドライブB：上に新しくMS-DOSシステムディスクを作成します(階層構造にMS-DOSシステムが必要なわけではなく、ここでの例にすぎない)。この作業の実行例は、今までに何度も登場していますので、ここでは省略しましょう。

【作業2】ルートディレクトリ上に、「CONFIG.SYS」ファイルと「PRINT.SYS」*ファイルをコピーし、３つの子ディレクトリを作成する

まず、大もとのディレクトリである「ルートディレクトリ」上に、MS-DOSシステム関係のファイルをドライブA：のシステムディスクからコピーします(ファイルを先にコピーしなければならない理由はなく、先にディレクトリを作成してもよいが)。セットしたドライブA：のディスクに目的のファイルがなければ、そのつど、別のディスクに交換して作業を進めてください。また流用できるCONFIG.SYSファイルがなければ、COPYコマンドなどで新たに作成します。

---

* MS-DOSのバージョン2.*x*では、「PRINT.SYS」に相当するプリンタを動かすためのプログラムはMS-DOSのIO.SYS部に組み込まれているので、これをコピーする必要はない。PC-9800シリーズのMS-DOSバージョン3.*x*では、そのプログラムは、デバイスドライバと呼ばれる外部のプログラムとして提供されるようになった。

ではファイルのコピーから始めましょう(図 11.4)。

```
A>COPY CONFIG.SYS B:⏎
        1 個のファイルをコピーしました.

A>COPY PRINT.SYS B:⏎
        1 個のファイルをコピーしました.

A>B:⏎

B>DIR⏎

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

COMMAND  COM    24931  88-07-13    0:00
CONFIG   SYS       87  89-07-29   17:25
PRINT    SYS     5855  88-07-13    0:00
        3 個のファイルがあります.
  1122304 バイトが使用可能です.

B>
```

ドライブA:のシステムディスクからドライブB:のディスクに2つのファイルをコピーする

カレントドライブをB:にチェンジする。今後の作業がしやすくなる

コピーの結果を確認する

作業1でコピーされたコマンド・プロセッサ

ルートディレクトリ上には3つのファイルがある

図 11.4　作業 2　ルートディレクトリ上に MS-DOS システム関係のファイルをコピーする

さてファイルをコピーしたら、次は容器である 3 つのディレクトリ、[WP] [MP] [MSDOS]を作成します。この作業により、3 つのファイルと、3 つのディレクトリがルートディレクトリ上に収容されることになります。ここで作成するディレクトリは、ルートディレクトリである**親ディレクトリ**に対して、**子ディレクトリ**に当たります。またこのような子ディレクトリのことを**サブディレクトリ**とも呼びます。

ディレクトリを作成するには MD コマンド(Make Directory)を使います。このコマンドの最も簡単な使い方は、

```
B>MD ディレクトリ名⏎
```

とキー入力することにより、現在のディレクトリ上に、[ディレクトリ名]という名前の子ディレクトリが作られます(図 11.5)。なお、作成するディレクトリの「名前」は自由ですが、使える文字や字数の制限などは、6.2 節で解説した、ファイル名の「主ファイル名」とまったく同じです(つまり字数は 8 文字まで)。

このコマンドの使い方は非常に簡単ですので、とりあえず実行してみますが、のちほど、改めて解説します。

```
B>MD WP⏎
                    ⎫
B>MD MP⏎            ⎬ MDコマンドによる3つのディレクトリの作成。ルートディレクトリ上に作成する
                    ⎭
B>MD MSDOS⏎

B>DIR⏎ ……………作成されたディレクトリを確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥

COMMAND  COM     24931  88-07-13   0:00 ⎫
CONFIG   SYS        87  89-07-29  17:25 ⎬ この3つはファイル
PRINT    SYS      5855  88-07-13   0:00 ⎭
WP           <DIR>      89-07-30  15:21 ⎫
MP           <DIR>      89-07-30  15:21 ⎬ 作成された3つのディレクトリ
MSDOS        <DIR>      89-07-30  15:21 ⎭
        6 個のファイルがあります.
  1119232 バイトが使用可能です.
              └── ファイルではなく、ディレクトリであることの表示
B>
          図11.3と同じものができた
```

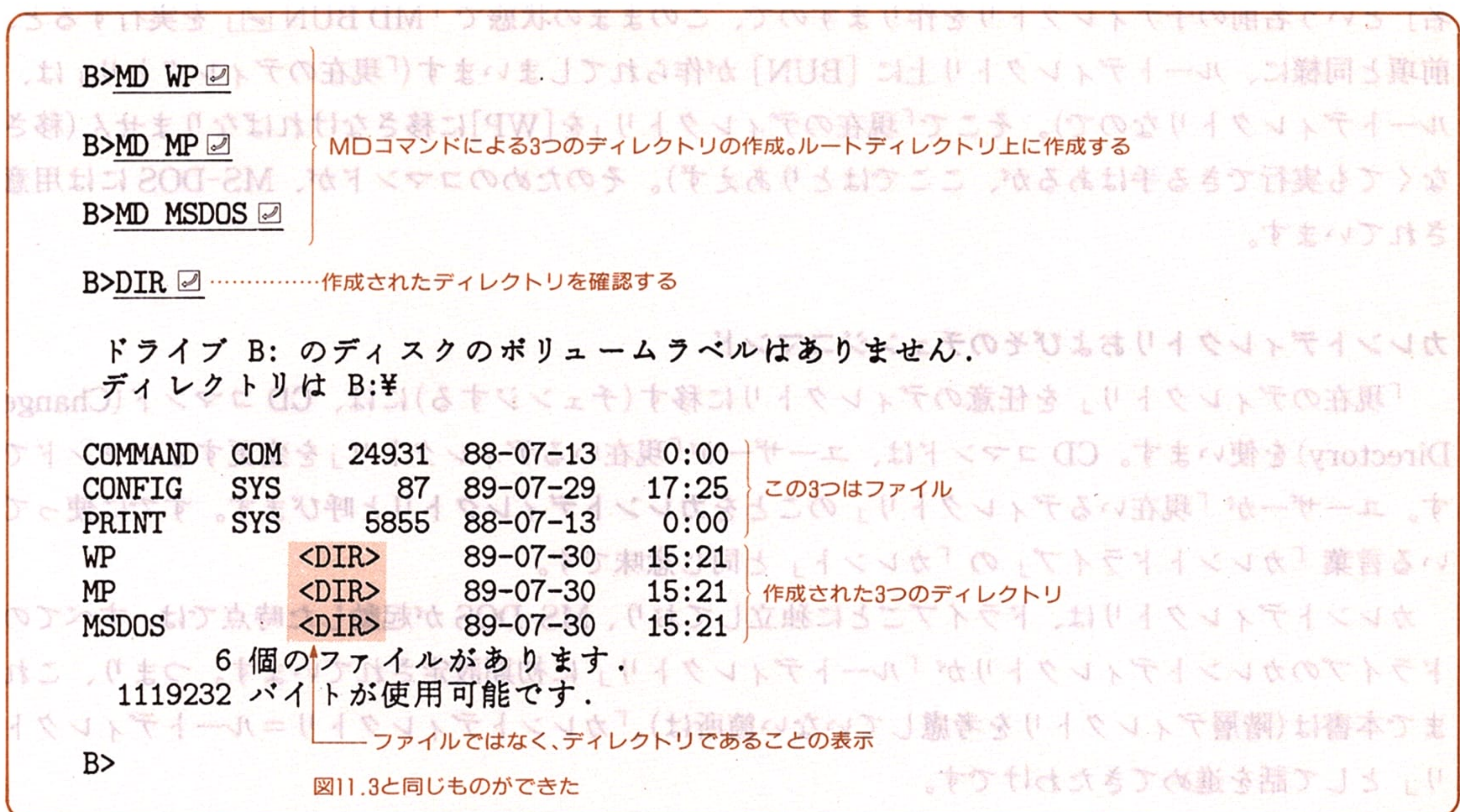
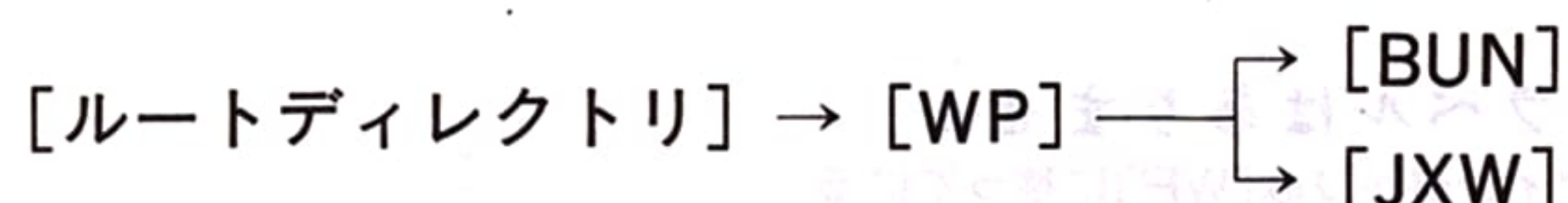



図 11.5　作業 2　ルートディレクトリ上に 3 つの子ディレクトリを作成する

【作業 3】ディレクトリ［WP］上に、2 つの子ディレクトリ［BUN］と［JXW］を作成し、それぞれの文書ファイルをコピーする

ディレクトリ［WP］上に、2 種類のワープロソフトの文書ファイルを収容するためのディレクトリ［BUN］と［JXW］を作成します。そして、それらのディレクトリ上に、それぞれの文書ファイルのいくつかをコピーします。

ここで作成するディレクトリ［BUN］は、親ディレクトリ［WP］の子ディレクトリとなりますが、親の［WP］は、ルートディレクトリの子ディレクトリですので、［BUN］はルートディレクトリから見ると「孫ディレクトリ」に当たります。このことは［JXW］についても同じです。結局ここでは、

```
                                    ┌→［BUN］
［ルートディレクトリ］→［WP］───┤
                                    └→［JXW］
```

という階層が作られるわけです。

この作業を行うには、**カレントディレクトリ**という概念の知識が必要になりますが、それについては実習の中で解説していきます。

ではまず、ディレクトリ［WP］上に子ディレクトリ［BUN］を作りましょう。さきに実行したディレクトリの作成コマンド「MD ディレクトリ名⏎」は、「現在のディレクトリ上」に、［ディレクトリ

名］という名前の子ディレクトリを作りますので、このままの状態で「MD BUN ⏎」を実行すると、前項と同様に、ルートディレクトリ上に［BUN］が作られてしまいます(「現在のディレクトリ」は、ルートディレクトリなので)。そこで「現在のディレクトリ」を［WP］に移さなければなりません(移さなくても実行できる手はあるが、ここではとりあえず)。そのためのコマンドが、MS-DOS には用意されています。

カレントディレクトリおよびそのチェンジコマンド

「現在のディレクトリ」を任意のディレクトリに移す(チェンジする)には、**CD** コマンド(Change Directory)を使います。CD コマンドは、ユーザーが「現在いるディレクトリ」を変更するコマンドです。ユーザーが「現在いるディレクトリ」のことを**カレントディレクトリ**と呼びます。すでに使っている言葉「カレントドライブ」の「カレント」と同じ意味です。

カレントディレクトリは、ドライブごとに独立しており、MS-DOS が起動した時点では、すべてのドライブのカレントディレクトリが「ルートディレクトリ」に初期設定されています。つまり、これまで本書は(階層ディレクトリを考慮していない箇所は)、「カレントディレクトリ＝ルートディレクトリ」として話を進めてきたわけです。

しかし階層ディレクトリ上でのファイルの操作は、カレントディレクトリをそのつど、適当なディレクトリにチェンジして(移して)作業を行うことが必要です。カレントディレクトリをチェンジしなくても操作する方法もありますが、慣れないうちは、CD コマンドを使って、カレントディレクトリを作業の対象となるディレクトリにチェンジしてから行う方がわかりやすく、確実な操作ができるでしょう。

では、カレントディレクトリを、現在のルートディレクトリ上の子ディレクトリに当たる［WP］にチェンジしてみましょう(図 11.6)。

```
B>CD WP⏎ ……………カレントディレクトリを、現在のディレクトリ上の[WP]に変更する

B>DIR⏎ ……………ディレクトリ[WP]を確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥WP ……………カレントディレクトリは[WP]に移っている

.            <DIR>      89-07-30   15:21 } MS-DOSの内部で自動的にこの[.]と[..]の
..           <DIR>      89-07-30   15:21 } 名前のディレクトリが作成される(後述)
        2 個のファイルがあります.
  1119232 バイトが使用可能です.

B>
```

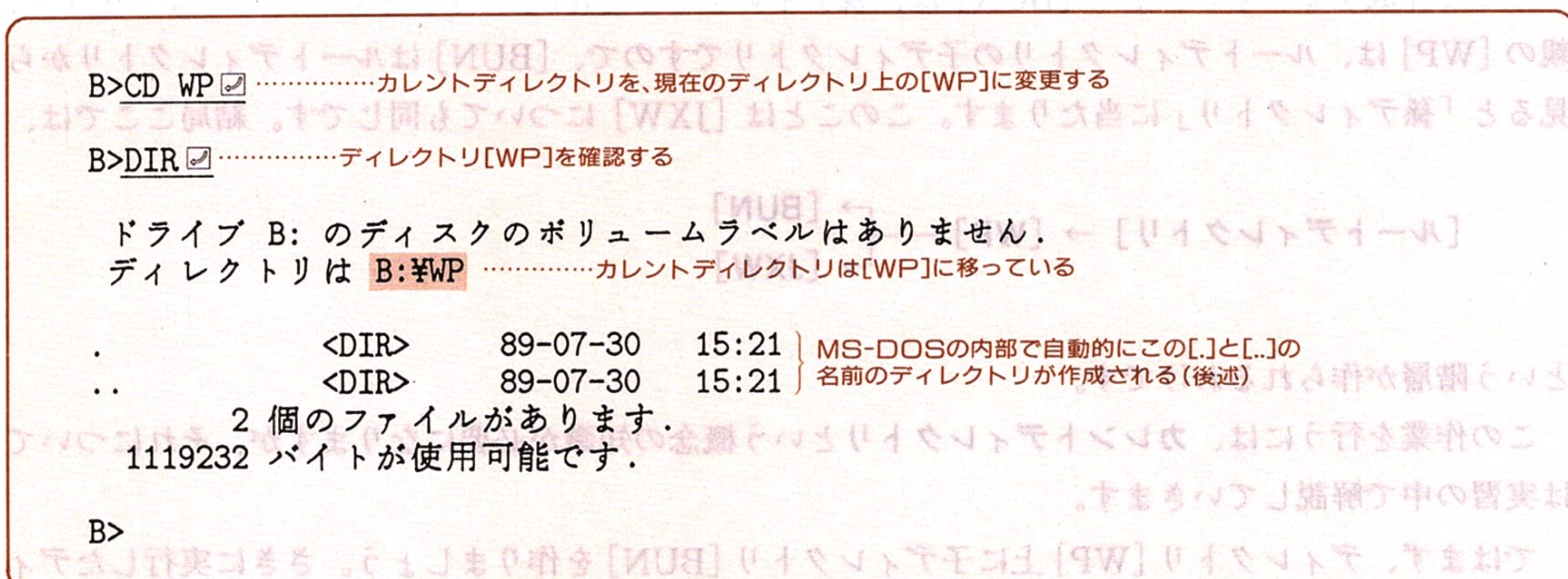



図 11.6　作業 3(a) カレントディレクトリをチェンジする

このように、カレントドライブが［WP］に移り、現在は、ディレクトリ［WP］上にいます。この［WP］上で実行するコマンドやプログラムは、ほかのディレクトリを指定しない限り、その影響が及ぶ範囲は、このディレクトリ［WP］内に直接存在するファイルに限られます。

つまり、たとえばここで全ファイルをすべて削除するコマンド「DEL ＊.＊⏎」を実行しても、ディレクトリ［WP］内に直接存在するファイル（現在このディレクトリ上にファイルはないが）が削除されるだけで、ルートディレクトリをはじめ、ほかのディレクトリ（自分の子ディレクトリを含めて）のファイルにはまったく影響はありません。この各ディレクトリ間の絶縁機能こそ、階層ディレクトリを利用してファイルを管理することの大きなメリットの１つなのです。

この実行例の、DIR コマンドの表示に新しく登場した記号、［.］と［..］の意味について、詳しくは後述しますが、とりあえず、［.］は「自分自身のディレクトリを表すディレクトリ名」、［..］は「自分の親ディレクトリを表すディレクトリ名」であると考えておいてください。

ではこれらの知識をもとに、サブディレクトリ［BUN］［JXW］を作成してみましょう（図 11.7）。

```
B>MD BUN ⏎     } カレントディレクトリ[WP]上に2つのディレクトリを作成する
B>MD JXW ⏎     }

B>DIR ⏎ ……………確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥WP ……………カレントディレクトリは[WP]

.            <DIR>      89-07-30    15:21    つまりカレントディレクトリ上のものしか
..           <DIR>      89-07-30    15:21    表示されていないことに注目
BUN          <DIR>      89-07-30    15:26  } 新たに作成された2つのディレクトリ
JXW          <DIR>      89-07-30    15:26  }
        4 個のファイルがあります.
  1117184 バイトが使用可能です.

B>
```

図 11.7　作業 3(b)　ディレクトリ［WP］上に、２つの子ディレクトリ［BUN］と［JXW］を作成する

以上の作業で、新松および一太郎の文書ファイルを収容するためのディレクトリ［BUN］と［JXW］が、その親ディレクトリとなる［WP］上に作成されました。

では次に、ディレクトリ［BUN］上に、ワープロソフト新松の文書ファイルのいくつかをコピーしましょう。階層的には次のような関係になります。

［ルートディレクトリ］→［WP］→［BUN］ーいくつかの文書ファイル

ではこの作業を実行しましょう。まずカレントディレクトリを現在の［WP］から、［BUN］にチェンジしてから COPY コマンドを実行します（図 11.8）。

```
B>CD BUN ⏎ ……………カレントディレクトリを[BUN]に移す

B>DIR ⏎ ……………確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥WP¥BUN ………カレントディレクトリは[BUN]に移った。
                          （このディレクトリの見方は本文11.2.2を参照）
.            <DIR>      89-07-30   15:26
..           <DIR>      89-07-30   15:26
        2 個のファイルがあります.
  1117184 バイトが使用可能です.

ドライブA:を新松の文書ディスクに入れ換えてから実行する

                  ┌─「このディレクトリ上に」という意味（「.」は省略してもよい）
B>COPY A:NMSD1 . ⏎ ……………ドライブA:上の文書ファイル「NMSD1」をドライブB:の「ここ」にコピーする
        1 個のファイルをコピーしました.

B>COPY A:NMSD2 . ⏎
        1 個のファイルをコピーしました.

B>COPY A:EIGYO5.BUN . ⏎
        1 個のファイルをコピーしました.
                                        同様に、4つの文書ファイルをコピーする
B>COPY A:EIGYO6.BUN . ⏎
        1 個のファイルをコピーしました.

B>COPY A:EIGYO7.BUN . ⏎
        1 個のファイルをコピーしました.

B>DIR ⏎ ……………確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥WP¥BUN ……………カレントディレクトリは[BUN]

.            <DIR>      89-07-30   15:26
..           <DIR>      89-07-30   15:26
NMSD1           14700   89-07-29   19:12
NMSD2           21455   89-07-21    1:27
EIGYO5   BUN    25609   89-07-29   18:59   ディレクトリ[BUN]上にコピーされた新松の文書ファイル
EIGYO6   BUN    12872   89-07-29   19:00
EIGYO7   BUN     8704   89-07-29   19:01
        7 個のファイルがあります.
  1031168 バイトが使用可能です.

B>
```

図 11.8　作業 3(c) ディレクトリ［BUN］上に、いくつかの文書ファイルをコピーする

ディレクトリ[JXW]上に一太郎の文書ファイルをコピーする実行例は、同じですので省略します。

【作業4】 ディレクトリ［MP］上に、Multiplanのいくつかのデータファイルをコピーする

ディレクトリ[MP]上に、表計算ソフト「Multiplan」のいくつかの必要なデータファイルをコピーします。階層的には次のような関係になります。

［ルートディレクトリ］→［MP］ーいくつかのデータファイル

この作業も、まずカレントディレクトリを[MP]に移してから――つまり「CD ¥MP⏎」を実行してから――COPYコマンドを実行しますが、同様ですので省略します。

【作業5】 ディレクトリ［MSDOS］上に、いくつかのMS-DOSコマンドのプログラムファイルコピーし、子ディレクトリ［TOOL］を作成する

まず最初に、ディレクトリ[MSDOS]上に、いくつかのMS-DOSの外部コマンドをコピーします。カレントディレクトリを[MSDOS]にチェンジし、COPYコマンドを使って、作業3と同様の方法で実行します。そのあとでディレクトリ[TOOL]を作成しましょう(図11.9)。

```
B>CD ¥MSDOS⏎ ……………カレントディレクトリを、ルートディレクトリ上のサブディレクトリ[MSDOS]に移す

B>DIR⏎ ……………確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥MSDOS ……………カレントディレクトリが[MSDOS]に移った

.            <DIR>      89-07-30   15:21
..           <DIR>      89-07-30   15:21
        2 個のファイルがあります.
   914432 バイトが使用可能です.

B>COPY A:FORMAT.EXE⏎
        1 個のファイルをコピーしました.

B>COPY A:SYS.EXE⏎
        1 個のファイルをコピーしました.

B>COPY A:DISKCOPY.EXE⏎
        1 個のファイルをコピーしました.

B>COPY A:CHKDSK.EXE⏎
        1 個のファイルをコピーしました.

B>COPY A:TREE.EXE⏎
        1 個のファイルをコピーしました.
```

（COPYコマンド群への注記）必要なファイルをコピーする。図11.8と同様であるが、ここではコピー先の「.」を省略している

— 図11.9 — (次ページに続く)

```
B>DIR↵ ……………確認する

 ドライブ B: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥MSDOS ……………カレントディレクトリは[MSDOS]

.            <DIR>      89-07-30  15:21
..           <DIR>      89-07-30  15:21
FORMAT   EXE    97766   88-07-13   0:00 ┐
SYS      EXE    25480   88-07-13   0:00 │
DISKCOPY EXE    19604   88-07-13   0:00 ├ ディレクトリ[MSDOS]上にコピーされたファイル
CHKDSK   EXE    10384   88-07-13   0:00 │
TREE     EXE     9608   88-07-13   0:00 ┘
        7 個のファイルがあります.
   748544 バイトが使用可能です.

B>MD TOOL↵ ……………サブディレクトリ[TOOL]を作成する

B>CD TOOL↵ ……………カレントディレクトリを[TOOL]にチェンジする

B>COPY A:EDLIN.EXE↵ ……………「EDLIN.EXE」をコピーする([TOOL]上にコピーされる)
        1 個のファイルをコピーしました.

B>DIR↵ ……………確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは B:¥MSDOS¥TOOL ……………カレントディレクトリは[TOOL]

.            <DIR>      89-08-03   15:05
..           <DIR>      89-08-03   15:05
EDLIN    EXE     7922   88-07-13    0:00 ……………ディレクトリ[TOOL]上にコピーされたファイル
        3 個のファイルがあります.
   749568 バイトが使用可能です.

B>
```

図 11.9　作業5ディレクトリ［MSDOS］上にいくつかの外部コマンドのプログラムファイルをコピーし、ディレクトリ［TOOL］を作成する

なお、ディレクトリ[TOOL]上には、MS-DOSのシステムディスクに含まれているエディタプログラムEDLIN.EXEをコピーしました。

以上の作業で、本章の最初に示した図11.2と同じ階層構造を持ったディスクができ上がりました。次節「11.3 階層ディレクトリの運用法」では、このディスクを使って、階層ディレクトリに関するさまざまな操作を実習してみましょう。

11.2.2 階層ディレクトリを操作するための「パス」と主要コマンド

さて、実習用のディスクができ上がり、一区切りしたところで、改めて階層ディレクトリに関する重要な機能や主要コマンドについて解説しておきましょう。もうすでに、今までの実習で使っている機能やコマンドもいくつかありますが、ここでは「パス」についての考え方、ディレクトリ作成コマンド「MD」、カレントディレクトリの変更コマンド「CD」、ディレクトリ削除コマンド「RD」、自分自身のディレクトリや自分の親ディレクトリを表す[.]や[. .]記号などにについて解説しましょう。

■ パスの考え方について

[本社]の→[総務部]の→[人事課]の－「鈴木さん」―― たとえ１万人の従業員がいる大会社であっても、このような具合に「鈴木さん」を指定すれば、簡単に、確実に見つけることができます。おそらく「鈴木さん」という人は、ほかにも何人かいると思われますが、「本社の→総務部の→人事課の」というように、目標の「鈴木さん」に至るまでの「経路」を示すことによって、１発で特定できたわけです。つまり、階層ディレクトリにおいては、目的のファイルに至るまでの、この「経路」のことが重要であり、そのことを**パス**と呼んでいます。

図 11.8 の実行例などに見られるディレクトリの書き方 ―― たとえば、

¥WP¥BUN

あるいはドライブ名を付けて、

B：¥WP¥JXW

のような表現が、「パス」の部分です。これはのちほど詳しく解説しますが、ルートディレクトリや、カレントディレクトリを出発点として、目的のディレクトリやファイルに至るまでの道順(経路)を書き表したものです。

パス名の一般形式は、対象とするドライブ名を x：とすると(カレントドライブの場合は省略できる)、カレントディレクトリから出発して、それに連なる(その子ディレクトリの方向の)目的のディレクトリを指す場合、

x：［子ディレクトリ名］¥［その子ディレクトリ名］……¥［目的のディレクトリ名］

となります。また目的のファイルを指す場合は、

x：［子ディレクトリ名］¥［その子ディレクトリ名］
……¥［その子ディレクトリ名］¥「目的のファイル名」

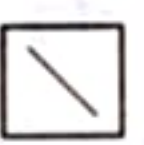
と記述します(「目的のファイル名」の頭に付けられた「¥」に注目)。

また、目的のディレクトリや目的のファイルが、自分の子ディレクトリの方向には存在しない場合は、いったんルートディレクトリに戻り、そこから出発して目的のディレクトリやファイルに向かわなければなりません。その場合、目的のディレクトリは、

x :¥［子ディレクトリ名］¥［その子ディレクトリ名］……¥［目的のディレクトリ名］

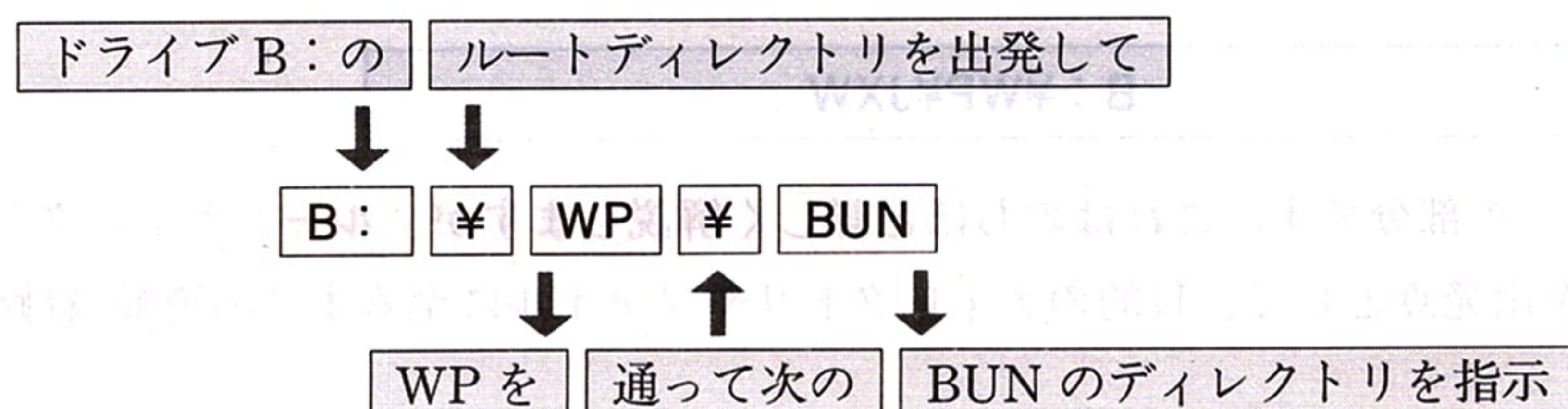
となります。また目的のファイルを指す場合は、

x :¥［子ディレクトリ名］¥［その子ディレクトリ名］
……¥［その子ディレクトリ名］¥「目的のファイル名」

と記述します(最初の「子ディレクトリ名」の頭に付けられた「¥」に注目)。

目的のパスを記述した部分を**パス名**と呼びますが、パス名における「¥」記号には、次の2つの意味があります。

- ドライブ名(省略される場合もあるが)の直後の「¥」―― つまり、いちばん最初の「子ディレクトリ」の頭に付けられたもの(波線部)―― は「ルートディレクトリ」を意味する
- ディレクトリとディレクトリとの間、およびディレクトリとファイル名との間の「¥」は、前後の経路をつなぎ合わせることを意味する

つまり、さきほどの例、「B:¥WP¥BUN」の意味は、

ドライブB:の　ルートディレクトリを出発して
B:　¥　WP　¥　BUN
WPを　通って次の　BUNのディレクトリを指示

という道順を示しています。どうして直接［BUN］と書いただけでは目的のディレクトリを指定できないかというと、親ディレクトリが異なれば、同じ名前のディレクトリがいくつ存在してもかまわないからです。もしかすると、ルートディレクトリ上や、［MP］上、［MSDOS］上、［TOOL］上にも、同じ名前のディレクトリ［BUN］が存在しているかも知れません。つまり、目的のディレクトリに至る経路 ―― パスを指定しなければ、目的のディレクトリを正しく特定することはできないのです。

なお、「¥」記号は、本来バックスラッシュの「＼」記号なのですが、私たちが使っている日本のパーソナル・コンピュータのキーボードでは、［＼］キーが［¥］キーに換えられているため、この唐突な「¥」記号を使わなければならないのです。

さて、これまでの実行例は、カレントディレクトリを、そのつど、それぞれの作業の対象となるディレクトリにチェンジしてから行っていました。この方法は、わかりやすく確実ではありますが、作業によってはカレントディレクトリをあちこちにチェンジせず、ある特定のディレクトリに固定したままで行いたい場合があります。そこで今後の実行例には、カレントディレクトリにこだわらず、パスを指定することによるファイルのコピーや、ディレクトリの作成などを行ってみましょう。

■ ディレクトリ作成コマンド　MD

ディレクトリを作成するには、MD コマンド(Make Directoty)を使います。このコマンドの使い方は、すでに実習していますが、ここでは、パスを指定して実行する一般的な話として、もう一度解説しておきましょう。

たとえば、これまでの例と同じ構造の階層ディレクトリにおいて、ドライブx：上の(カレントドライブの場合、x：は省略可)ディレクトリ[MSDOS]上に子ディレクトリ[TOOL]を作成する場合、一般にはルートディレクトリから出発するパスを指定して、

```
B>MD x：¥MSDOS¥TOOL↵
```

［ルート］から出発し→［MSDOS］上に→［TOOL］を作成する

となります。この場合は、カレントディレクトリがどこであっても実行することができます。もし、カレントディレクトリが［ルート］であれば、最初の「¥」は不要です。

また、カレントディレクトリが[MSDOS]であれば、自分の直接の子ディレクトリを作成することになるので、パスを指定する必要はなく、

```
B>MD x：TOOL↵
```

だけで実行できます(さきほど実習したのはこの形式)。

作成するディレクトリ名に使える文字や字数の制限は、6.2 節で解説したファイル名の主ファイル名と同じです(文字数は 8 文字まで)。またディレクトリ名は、直接の親ディレクトリが異なれば、ほかに同じ名前のディレクトリが、どこにいくつ存在してもかまいません。［総務］の鈴木さん、［人事］の鈴木さん、［営業］の鈴木さん、…のように、部や課が異なれば、同じ名前の「鈴木さん」が何人いてもよく、それらの区別もできるわけです。ここの例であれば、ルートディレクトリ上にも、［MP］上、［WP］上、［BUN］上、［JXW］上にも、同じ名前のディレクトリ［TOOL］を作ることができます。同様に、ファイルに関しても、収容する直接のディレクトリが異なれば、同じ名前のファイルが、同一のディスク上にいくつあってもかまいません。

■ カレントディレクトリ変更コマンド　CD

階層ディレクトリを利用してさまざまな作業をしている場合、ユーザーは必ずいずれかのディレクトリに居ます。その「今居る」ディレクトリ —— つまりカレントディレクトリをチェンジするコマンドがCDコマンド(Change Directoty)です。このコマンドの使い方は、すでに実習していますが、ここでは、パスを指定して実行する一般的な話として、もう一度解説しておきましょう。

実習の例にしている階層ディレクトリにおいて、ドライブx：上の(カレントドライブの場合、x：は省略可)カレントディレクトリを[TOOL]に移動(チェンジ)するには、一般的にはルートディレクトリから出発するパスを指定して、

```
B>CD x:¥MSDOS¥TOOL ⏎
```

[ルート] から出発し→ [MSDOS] を経由して→ [TOOL] に移る

となります。この場合は、カレントディレクトリがどこであってもよいわけです。もし、カレントディレクトリが［ルート］であれば、最初の「¥」は不要です。

また、カレントディレクトリが[MSDOS]であれば、自分の直接の子ディレクトリに移ることになるので、パスを指定する必要はなく、

```
B>CD x:TOOL ⏎
```

だけでもかまいません(さきほど実習したのはこの形式)。

■ ディレクトリ削除コマンド　RD

ディレクトリを削除するコマンドは、今までの実習では登場していません。ここで初めて登場します。不要になったディレクトリを削除するには、RDコマンド(Remove Directory)を使います。たとえば、ドライブx：上の(カレントドライブの場合、x：は省略可)ディレクトリ[MSDOS]の子ディレクトリ[TOOL]を削除する場合は、一般にはルートディレクトリから出発するパスを指定して、

```
B>RD x:¥MSDOS¥TOOL ⏎
```

[ルート] から出発し→ [MSDOS] 上の→ [TOOL] を削除する

となります。この場合は、カレントディレクトリがどこであってもよいわけです。もし、カレントディレクトリが［ルート］であれば、最初の「¥」は不要です。

また、カレントディレクトリが[MSDOS]であれば、削除するディレクトリが自分の直接の子ディレクトリに当たるため、パスを指定する必要はなく、

```
B>RD x:TOOL ⏎
```

だけでも実行できます。ただしいずれの場合も、何らかのファイルや、子ディレクトリが収容されているディレクトリは削除することができません。また、カレントディレクトリ(つまり自分自身)も削除することができません。というのは、親ディレクトリが消滅すれば、それに収容されているすべてのファイルや子ディレクトリが宙に浮いてしまいます。また、カレントディレクトリが消滅すれば、自分の居場所がなくなるというわけです。したがって、不要なディレクトリを削除するには、まずそのディレクトリに連なる、すべてのファイルやディレクトリを削除して、さらにカレントディレクトリをその親側のいずれか、もしくは関係のないディレクトリに移してから RD コマンドを実行しなければなりません。

■ 自分自身と、親ディレクトリを表す代用記号　[.] [..]

ディレクトリ作成コマンド「MD」を実行し、新しいディレクトリを作成すると、そのディレクトリ上に必ず2つの「ディレクトリ」[.] と [..] が作成されています。これは、ある1つのディレクトリが作成されるとき、MS-DOS が自動的に作成する2種類の「ディレクトリ」であり、それぞれ次のような意味を持っています。

[.] は、自分自身のディレクトリ――つまり、カレントディレクトリ自身を表す。たとえば、カレントディレクトリが [BUN] であれば、

[.]＝[BUN]

として、各種のコマンドに代用できる。

また [..] は、「親ディレクトリ」――つまり、自分を収容しているディレクトリを表し、たとえば、カレントディレクトリが [BUN] であれば、[WP] ← [BUN] の関係なので、親ディレクトリは [WP] であり、

[..]＝[WP]

として、各種のコマンドに代用できる。またカレントディレクトリが [WP] の場合であれば、[ルート] ← [WP] の関係なので、

[..]＝[ルートディレクトリ]

として、各種のコマンドに代用できる。

したがって、カレントディレクトリが［BUN］のとき、CD コマンドを、

CD ..¥..⏎

のように実行すれば、カレントディレクトリは、

［ルート］ ← ［WP］ ← ［BUN］

のように、2 代前の親ディレクトリ —— つまりルートディレクトリに移ることになります。つまり、[.]や[..]記号は、パス名やディレクトリ名を絶対的に指定するのではなく、カレントディレクトリを基準にして、「相対的に目的のディレクトリを指定することができる記号」なのです。

では、これまでの基礎知識をもとに、次節では、階層ディレクトリのいろいろな操作を行ってみることにしましょう。

11.3 階層ディレクトリの運用法

さて、これまでの作業で、ドライブ B：には図 11.2 に示した階層ディレクトリと同じ構成のディスクが作成されています。本節では、このディスク上でいろいろな操作を行い、階層ディレクトリの運用法を実習しましょう。ただし、ここでは基本的な操作法を経験することが主眼であり、解説もそれに沿った基礎的なものです。実行例によく注目して、それぞれ何を行っているのか、その意味をよく汲み取るようにしてください。また、図 11.2 と対比して、現在自分が居るディレクトリ(カレントディレクトリ)や、操作の対象とする目的のディレクトリ、また、目的のファイルが収容されているディレクトリとの相互関係、その指定法などに注目してください。そこが本節のキーポイントです。

作成したドライブ B：上のディスクには、MS-DOS システムがコピーされていますので、これをドライブ A：にセットして MS-DOS を起動し、ドライブ A：上でいろいろな操作を行ってみましょう(今までどおり、ドライブ B：のままでもかまわないが)。

まず、自分が今どこのディレクトリにいるか —— つまりカレントディレクトリがどこであるのかを調べるコマンド(パラメータを付けない CD コマンド)の実習と、カレントディレクトリをチェンジする実習です。CD コマンドは、すでに前節で使っていますが、内蔵コマンドですので、そのコマンドのプログラムファイルが存在するドライブやディレクトリを考慮することなく実行できます。

```
A>CD↵ ……………このようにパラメータを付けずに実行すると、カレントディレクトリを表示する
A:¥MSDOS ………現在はルートディレクトリ上の[MSDOS]であった

A>CD ¥WP¥BUN↵ ……………いったんルートディレクトリに戻って、[WP]上の[BUN]にディレクトリを移す

A>CD↵ ……………確認する
A:¥WP¥BUN ………確かに[BUN]に移っている

A>CD ..↵ ……………親ディレクトリに移動する

A>CD↵ …………確認する
A:¥WP ……………[BUN]の親ディレクトリである[WP]に移動している

A>CD ..¥MSDOS↵ ……………さらに親ディレクトリ(つまりルートディレクトリ)に移り、その上のサブディレクトリ
                           [MSDOS]に移動する

A>CD↵ ……………確認する
A:¥MSDOS ………確かに[MSDOS]に移っている

A>CD TOOL↵ ……………カレントディレクトリ上のサブディレクトリ[TOOL]にチェンジする(¥を付けていないことに注目)

A>CD↵ ………………………確認する
A:¥MSDOS¥TOOL………確かに[TOOL]に移っている

A>
```

図 11.10　CD コマンドを使ってカレントディレクトリがどこかを調べたり、チェンジしたりする

次にディレクトリ［BUN］上のワープロの文書ファイル「NMSD1」(これはテキストファイル形式)を、TYPE コマンドで画面に表示してみましょう。TYPE コマンドは内蔵コマンドですので、そのコマンドのプログラムファイルが存在するドライブやディレクトリを考慮することなく実行できます。

目的のファイル「NMSD1」は、ルートディレクトリから見ると、次のようなパスをたどります(*印はカレントディレクトリを示す)。

［ル*ート］→［WP］→［BUN］−NMSD1

したがって、現在のカレントディレクトリ［ルート］から、「NMSD1」が存在するディレクトリに到達するためのパスの指定は、

¥WP¥BUN

あるいは「¥」を省略して、

WP¥BUN

という具合になります([WP]はカレントディレクトリの子ディレクトリなので、いったんルートディレクトリに戻る指示である最初の「¥」は必要がない)。

では、カレントディレクトリが[ルート]の場合の、TYPEコマンドを使って[BUN]上のファイル、「NMSD1」を画面に表示する実行例を示しましょう(図11.11)。

```
A>CD⏎ ………カレントディレクトリの確認
A:¥ ……………ルートディレクトリである

A>TYPE WP¥BUN¥NMSD1⏎ ……………ディレクトリ[WP]→[BUN]上の文書ファイル「NMSD1」を表示する

1章　パソコン・ユーザーとMS-DOSとのかかわり

　ソフトウェア・シーンは今、MS-DOSの時代

　現在の最も一般的なパーソナル・コンピュータである、16ビット・マシン上で使われている実務ソフトのほとんどすべては、「 MS-DOS」(ﾙﾋﾞ: ｴﾑｴｽﾄﾞｽ)という「 基本ソフトウェア」( Operating System: OS) ( ﾙﾋﾞ: ｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑ／ｵｰｴｽ) 上で動作するように作られています。本格的な実務ソフトは、BASIC( プログラミングの入門用言語の1つ) の世界
…OSを基盤とする世界です。
面的なものはCP/Mの体系を受け継いで…一般のビジネスソフト利用…
る内部構造は、CP/Mとは大きく異なり、非常に強力になりました。…ディスクファイルの互換性はありません。
　ただし、.システムコールの基本的なものについては、CP/Mとの互換性が考慮されていますので、CP/M用に作られたソフトウェアを、MS-DOS上に移植することが、ある程度は可能になっています( ソースレベルでの話)。

A>
```

図11.11　ディレクトリ［BUN］上の文書ファイルをタイプアウトする

もし、カレントディレクトリが［JXW］であった場合、同じことを実行する際のパスは、次に示すように、いったん親ディレクトリの[WP]、あるいはルートディレクトリに戻ってから目的の[BUN]に向かって出発しなければなりません(*印はカレントディレクトリを示す)。

```
                         ↗ [BUN]— NMSD1
親ディレクトリ [WP]
                         ↖ [JXW]
                               *
```

あるいは、

```
[ルート] → [WP] → [BUN]—NMSD1
↑——————————————— [JXW]
                      *
```

したがって、目的のファイル、「NMSD1」が存在するディレクトリに到達するための「パス」は、

..¥BUN

（共通の親ディレクトリまで戻る場合）

あるいは、

¥WP¥BUN

（ルートディレクトリまで戻る場合。「ルートディレクトリに戻り、そこから出発する」という意味であり、先頭の「¥」は絶対に必要）

となります。

ではこの2種類のパス指定によるTYPEコマンドの実行例を示しましょう（図11.12）。

```
A>CD ……………カレントディレクトリを確認する
A:¥WP¥JXW ………［JXW］である

A>TYPE ..¥BUN¥NMSD1 ……………［JXW］の親ディレクトリに戻り、その上の［BUN］上の文書ファイル
                               「NMSD1」を表示する

1章　パソコン・ユーザーとMS-DOSとのかかわり

  ソフトウェア・シーンは今、MS-DOSの時代

  現在の最も一般的なパーソナル・コンピュータである、16ビット・マシン上で使われて
いる実務ソフトのほとんどすべては、「 MS-DOS」（ﾙﾋﾞ： ｴﾑｴｽﾄﾞｽ）という「 基本ソフトウェ
ア」（ Operating System: OS）（ ﾙﾋﾞ： ｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑ／ｵｰｴｽ）上で動作するように作ら
れています。本格的な実務ソフトは、BASIC（ プログラミングの入門用言語の1つ）の世界
とは関係のない、MS-DOSで代表されるOSを基盤とする世界です。
  ソフトウェアの開発に携わる専門家は当然のことながら、一般のビジネスソフト利用者
にとっても、OSの知識は非常に重要です。つまり、コンピュータを自在に活用するために
は、OSの基礎知識がぜひとも必要なのです。
  まず本章では、MS-DOSというOSが、何をする場合にも「 基本ソフトウェア」として存在
することを認識するとともに、OSがなぜ必要なのかを解説します。
                                   今すぐ理解する必要はありません。そう簡
～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～
  両者にはこのような相違があるにしても、MS-DOSは、コマンドの名前や操作法などの表
面的なものはCP/Mの体系を受け継いでいます。しかし、ファイル・システムをはじめとす
る内部構造は、CP/Mとは大きく異なり、非常に強力になりました。そのため、ディスクフ
ァイルの互換性はありません。
  ただし、システムコールの基本的なものについては、CP/Mとの互換性が考慮されていま
すので、CP/M用に作られたソフトウェアを、MS-DOS上に移植することが、ある程度は可能
になっています（ ソースレベルでの話）。

A>
```

—図11.12—（次ページに続く）

```
A>TYPE ¥WP¥BUN¥NMSD1⏎
```
いったんルートディレクトリに戻り、[ルート]→[WP]→[BUN]上の文書ファイル「NMSD1」を表示する

```
1章　パソコン・ユーザーとMS-DOSとのかかわり

  ソフトウェア・シーンは今、MS-DOSの時代

  現在の最も一般的なパーソナル・コンピュータである、16ビット・マシン上で使われて
いる実務ソフトのほとんどすべては、「 MS-DOS」(ﾙﾋﾞ: ｴﾑｴｽﾄﾞｽ)という「 基本ソフトウェ
ア」( Operating System: OS) ( ﾙﾋﾞ:  ｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑ／ｵｰｴｽ) 上で動作するように作ら
れています。本格的な実務ソフトは、BASIC( プログラミングの入門用言語の1つ) の世界
とは関係のない、MS-DOSで代表されるOSを基盤とする世界です。
  ソフトウェアの開発に携わる専門家は当然のことながら、一般のビジネスソフト利用者
にとっても、OSの知識は非常に重要です。つまり、コンピュータを自在に活用するために
は、OSの基礎知識がぜひとも必要なのです。
  まず本章では、MS-DOSというOSが、何をする場合にも「 基本ソフトウェア」として存在
                                     のかを解説します。
面的なものはCP/Mの体系を受け継いでいます。
る内部構造は、CP/Mとは大きく異なり、非常に強力になりました。そのため、ディスクフ
ァイルの互換性はありません。
  ただし、システムコールの基本的なものについては、CP/Mとの互換性が考慮されていま
すので、CP/M用に作られたソフトウェアを、MS-DOS上に移植することが、ある程度は可能
になっています( ソースレベルでの話)。

A>
```

図 11.12　パスを指定する方法の違いによる２種類のTYPEコマンドの実行

次に、ディレクトリ［MSDOS］上のCHKDSKコマンドを実行してみましょう。CHKDSKコマンドは外部コマンドであり、今までの内蔵コマンドと違い、実行するには、そのプログラムファイルが、どのディレクトリに存在するかが問題になります。

現在のカレントディレクトリは、さきの実行例のままのルートディレクトリですので、このままでは実行できません。実行するためには、カレントディレクトリを、目的のプログラムファイル（ここではCHKDSK.EXE（あるいは.COM））が存在するディレクトリにチェンジするか、あるいは、カレントディレクトリから目的のプログラムファイルに至るまでのパスを指定するかしなければなりません。ただし、MS-DOSのバージョン3.xでは、パスを指定することにより、直接任意のプログラムの実行が可能です（後述）。

まずここでは、カレントディレクトリを目的のプログラムファイル「CHKDSK.EXE」が存在するディレクトリにチェンジして、CHKDSKコマンドを実行してみましょう。CHKDSKコマンドを実行するためのプログラムファイル「CHKDSK.EXE」（あるいは.COM）は、次に示すように、ディレクトリ［MSDOS］上に存在しています（*印はカレントディレクトリ）。

［ルート*］→［MSDOS］－CHKDSK.EXE

では、カレントディレクトリを[MSDOS]に移してから、ディスク上のすべてのディレクトリとファイルを表示する、/V スイッチ付きの CHKDSK コマンドを実行してみましょう(図 11.13)。

```
A>CD ¥MSDOS ……………カレントディレクトリをルートディレクトリ上のサブディレクトリ[MSDOS]に移す

A>CD ……………カレントディレクトリを確認する
A:¥MSDOS ………[MSDOS]に移っている

A>CHKDSK/V ……………/Vスイッチを付けて CHKDSKコマンドを実行する
ディレクトリ  A:¥ ……ルートディレクトリに収容されているすべてのディレクトリとファイルの状態が表示される
        A:¥IO.SYS
        A:¥MSDOS.SYS
        A:¥COMMAND.COM          ルートディレクトリ上のファイル
        A:¥CONFIG.SYS
        A:¥PRINT.SYS
ディレクトリ  A:¥WP
ディレクトリ  A:¥WP¥BUN
        A:¥WP¥BUN¥NMSD1
        A:¥WP¥BUN¥NMSD2         ルートディレクトリ上のサブディレクトリ[WP]に連なる
        A:¥WP¥BUN¥EIGYO5.BUN    ディレクトリとファイル
        A:¥WP¥BUN¥EIGYO6.BUN
        A:¥WP¥BUN¥EIGYO7.BUN
ディレクトリ  A:¥WP¥JXW
ディレクトリ  A:¥MP
        A:¥MP¥URIAGE.MP
        A:¥MP¥MACHINE.MP        ルートディレクトリ上のサブディレクトリ[MP]に連なるファイル
        A:¥MP¥MITUMORI.MP
ディレクトリ  A:¥MSDOS
        A:¥MSDOS¥FORMAT.EXE
        A:¥MSDOS¥SYS.EXE
        A:¥MSDOS¥DISKCOPY.EXE   ルートディレクトリ上のサブディレクトリ[MSDOS]に連なる
        A:¥MSDOS¥CHKDSK.EXE     ファイルとディレクトリ
        A:¥MSDOS¥TREE.EXE
ディレクトリ  A:¥MSDOS¥TOOL
        A:¥MSDOS¥TOOL¥EDLIN.EXE

  1250304 バイト : 全ディスク容量
    95232 バイト : 2 個のシステムファイル
     6144 バイト : 6 個のディレクトリ
   409600 バイト : 17 個のユーザーファイル
   739328 バイト : 使用可能ディスク容量

   655360 バイト : 全メモリ
   454096 バイト : 使用可能メモリ

A>
```

図 11.13　実行するプログラムが存在するディレクトリにカレントディレクトリをチェンジして、目的の外部コマンドを実行する

次に、カレントディレクトリが［BUN］のとき、CHKDSKコマンドのプログラムファイルが存在するディレクトリに至るパスを指定することにより、このコマンドを実行してみましょう。ただし、さきにも述べたように、パスの指定により実行できるのは、MS-DOSのバージョン3.xからの機能です。カレントディレクトリが［BUN］ですので、ディレクトリ［MSDOS］上に存在する目的のファイル「CHKDSK.EXE」に到達するには、次に示すように、いったんルートディレクトリに戻ってから目的のディレクトリに向かって出発しなければなりません。

```
 ┌──────[BUN]*
 ↓
[ルート] → [MSDOS]─ CHKDSK.EXE
```

したがって、「CHKDSK.EXE」が存在するディレクトリに到達するための「パス」は、

¥MSDOS

となります。先頭の「¥」は絶対に必要です。この「¥」は、「ルートディレクトリに戻り、そこから出発する」という重要な意味があります。

```
A>CD
A:¥WP¥BUN ……………カレントディレクトリは[WP]上の[BUN]である

A>¥MSDOS¥CHKDSK ……………ルートディレクトリ上のサブディレクトリ[MSDOS]にある
                           CHKDSKコマンドをパス名を指定して直接実行する

  1250304 バイト : 全ディスク容量
    95232 バイト : 2 個のシステムファイル
     6144 バイト : 6 個のディレクトリ
   409600 バイト : 17 個のユーザーファイル
   739328 バイト : 使用可能ディスク容量          実行結果

   655360 バイト : 全メモリ
   454096 バイト : 使用可能メモリ

A>          (このようなパス名を指定しての実行は、MS-DOSのバージョン3.x以降で可能)
```

図 11.14　パス名を指定して目的の外部コマンドを実行する

次に、階層ディレクトリの構造や、その内容を表示するための専用コマンド**TREE**コマンドを使って、実習で作成したディスクを調べてみましょう。前項のCHKDSKコマンドと違い、ディスクの内容がディレクトリごとに整理され、わかりやすく表示されます(**図 11.15**)。なお、このコマンドは、MS-DOSのバージョン3.xのシステムディスクに付属している外部コマンドであり、実行するにはプログラムファイル「TREE.EXE」が必要です。今回はカレントディレクトリ上で実行しましょう。

(あらかじめ、カレントディレクトリを[MSDOS]にチェンジしておく)

A>TREE/F ⏎ ……………カレントディレクトリ[MSDOS]上のTREEコマンドを実行する。
スイッチ「/F」により、各ディレクトリ上に存在するファイル名も表示される

```
ディレクトリ・パス・リスト

ファイル:            COMMAND  .COM
                     CONFIG   .SYS
                     PRINT    .SYS
```

ルートディレクトリ上のファイル

```
パス: A:¥WP

サブディレクトリ:    BUN
                     JXW

ファイル:            なし
```

[WP]上の状態

```
パス: A:¥WP¥BUN

サブディレクトリ:    なし

ファイル:            NMSD1
                     NMSD2
                     EIGYO5   .BUN
                     EIGYO6   .BUN
                     EIGYO7   .BUN
```

[BUN]上の状態

```
パス: A:¥WP¥JXW

サブディレクトリ:    なし

ファイル:            なし
```

[JXW]上の状態

```
パス: A:¥MP

サブディレクトリ:    なし

ファイル:            URIAGE   .MP
                     MACHINE  .MP
                     MITUMORI .MP
```

[MP]上の状態

— 図 11.15 — (次ページに続く)

```
パス: A:¥MSDOS

サブディレクトリ: TOOL

ファイル:          FORMAT  .EXE
                   SYS     .EXE
                   DISKCOPY.EXE
                   CHKDSK  .EXE
                   TREE    .EXE
```

[MSDOS]上の状態

```
パス: A:¥MSDOS¥TOOL

サブディレクトリ: なし

ファイル:          EDLIN   .EXE
```

[TOOL]上の状態

```
A>
```

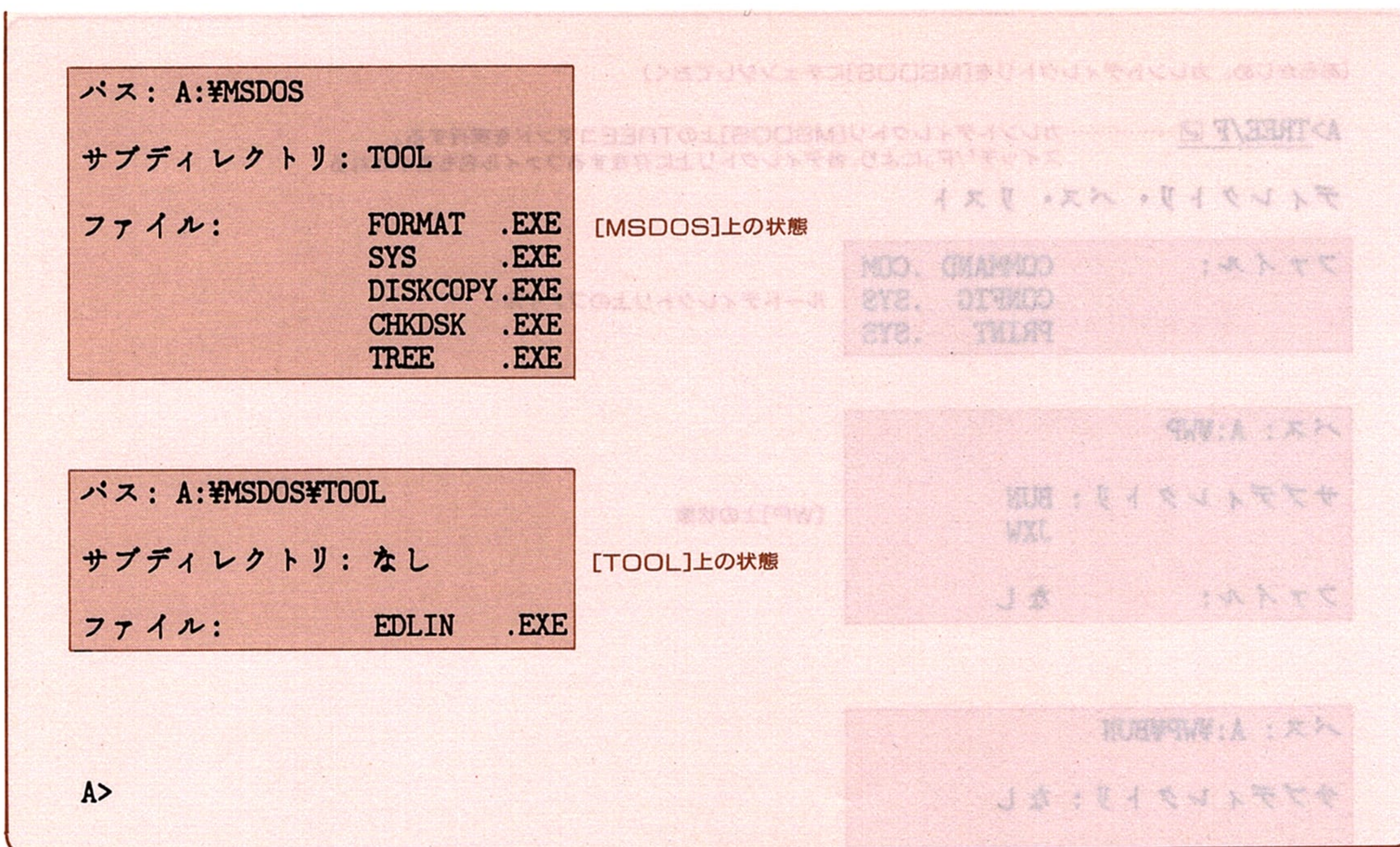



図 11.15　階層構造やその内容を表示する専用コマンド「TREE」を使って全ディレクトリの状況を調べる

さて次は **PATH** コマンドの実習です。

階層ディレクトリを利用している場合、今までの実行例からもわかるように、外部コマンド(外部プログラム)は、カレントディレクトリ上のものが実行されます。つまり、よそのディレクトリ上のプログラムは、MS-DOS のバージョン 2.*x* の場合、パスを指定しても実行することができません。バージョン 3.*x* の場合は、パスを指定すれば実行することが可能ですが、いちいちパス名を付けるのもめんどうです。そこで MS-DOS には、**PATH** コマンドが設けられています。PATH コマンドを使って、

A>PATH　*x*：ディレクトリ名１；*y*：ディレクトリ名２；…

のように、任意のドライブ名やディレクトリ名を指定しておくことにより、指定されたドライブの指定されたディレクトリ上のプログラムは、カレントドライブやカレントディレクトリに関係なく、またそれらのパスを指定することなく実行することができるのです(複数のディレクトリを指定する場合は、「;」で区切る。カレントドライブの場合はドライブ名 *x*：を省略できる)。このように PATH コマンドで、任意のドライブ上の必要なディレクトリを指定することを、本書では **PATH 設定**と呼ぶことにします(「パスを切る」などともいう)。

PATH コマンドのその他の機能としては、次のようなものがあります。

A>x：PATH ⏎………ドライブx：上のPATH設定の状況を表示する
A>x：PATH；⏎………ドライブx：上に設定してあるパスをすべて解除する

では、PATH設定を［MSDOS］と［TOOL］の2つのディレクトリに設定して、そこに収容されているいくつかのコマンドを実行してみましょう(図11.16)。カレントディレクトリはどこであっても関係ありません。なお、ディレクトリ［TOOL］上には「EDLIN」がコピーされているとしましょう。この「EDLIN」は、MS-DOSのシステムディスクに標準装備のエディタ(古典的なラインエディタの形式のもの)ですが、現在では各社から優れたスクリーンエディタが発売されているため、一般にはほとんど使われない存在になりました。EDLINの使い方は解説していませんが、ここでは誰でもできる非常に簡単な操作しか行いませんので安心してください。

```
A>PATH ¥MSDOS;¥MSDOS¥TOOL ⏎ ……「パス」をカレントドライブ(A：)のルートディレクトリ上の[MSDOS]と、
                              その上の[TOOL]の2つのディレクトリに設定する

A>PATH ⏎ ……パラメータを付けずに実行すると、現在の「パス」を表示する
PATH=¥MSDOS;¥MSDOS¥TOOL ……「パス」は[MSDOS]と[TOOL]の2つのディレクトリ

A>CD ⏎ ……カレントディレクトリを確認してみる
A:¥ ……ルートディレクトリにいる

A>EDLIN WP¥TEST1 ⏎ ……[TOOL]上にある「EDLIN.EXE」を実行して、サブディレクトリ
                       [WP]上に「TEST1」ファイルを作成する
新しいファイルです. ……EDLINが起動された
*I ⏎ ……INSERTコマンドでインサートモードにはいる
      1:*階層ディレクトリの実習　パスを設定して、WPにファイルTEST1を作ります。⏎
      2:*^Z ⏎ ……CTRL+Zでインサートモードを終わる。「^」+「Z」ではなく CTRL+Z を入力する
*E ⏎ ……ENDコマンドでEDLINを終了する。ディレクトリ[WP]上に
        「TEST1」というファイルが作成される
                              日本語入力機能で文章を入
                              力した。めんどうな人は、通
                              常入力の「ABCDEFG⏎」
                              とでも入力しておく
A>DIR WP ⏎ ……ディレクトリ[WP]を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥WP

.            <DIR>      89-07-30   15:21
..           <DIR>      89-07-30   15:21
BUN          <DIR>      89-07-30   15:26
JXW          <DIR>      89-07-30   15:26
TEST1              72   89-07-30   16:58 ……EDLINで作成されたファイル
         5 個のファイルがあります.
   748544 バイトが使用可能です.

A>
```

図11.16 PATH設定によるカレントディレクトリ以外のプログラムの実行

このように、実行するプログラムが収容されているドライブやディレクトリを、PATHコマンドであらかじめ指定しておけば、カレントドライブやカレントディレクトリに関係なく目的のプログラムを起動することができます。つまり、実行を指示されたプログラムファイルを捜す範囲が、カレントドライブ上のカレントディレクトリだけでなく、PATHコマンドで設定されたすべてのディレクトリに及ぶわけです。

なお、このようなPATH設定のためのコマンドを、自動スタートバッチファイル(AUTOEXEC.BAT)に記述しておけば、MS-DOSを起動した時点で、自動的にPATH設定が行われることになります。多くの場合、PATH設定は自動スタートバッチファイルで行われます。

ただしPATH設定を行う意味は、コマンドやプログラムの実行を、カレントディレクトリに関係なく行うことであり、そのプログラムの実行によって処理しようとするファイルの指定には、やはりパスの指定が必要です。たとえばここの実行例では、「¥WP¥TEST1」と指定することにより、ディレクトリ[WP]上に新しいファイルを作成しました。

さて次は、PATHコマンドを使わずに、カレントディレクトリ以外のディレクトリ上のプログラムを、ダイレクトに実行してみましょう。さきの図11.14でも行っていますが、このダイレクトに実行できる機能は、MS-DOSのバージョン2.*x*にはありません。バージョン3.*x*でのみ可能です。

従来、階層ディレクトリ上のプログラムを実行するには、次の2つの方法がありました。

1) 実行しようとするプログラムが収容されているディレクトリに、カレントディレクトリを移す
2) PATHコマンドを使って、あらかじめそのディレクトリにPATHを設定しておく

ところがバージョン3.*x*以降、プログラムを実行する機能が拡張され、一般的なファイルの指定と同様に、実行するプログラムファイルにパス名を付ければ、カレントディレクトリに関係なく直接実行することができるようになりました。

たとえば、前項と同様、ディレクトリ［TOOL］上のエディタプログラム「EDLIN」を実行して、ファイル「ファイル名」を作成するには、カレントディレクトリが現在どこであろうと関係なく、

```
A>x：¥MSDOS¥TOOL¥EDLIN ファイル名↵
```

とキー入力すればよいわけです。つまり、「¥MSDOS¥TOOL¥」という、「EDLIN.EXE」に至るパス名を付けて実行すればよいのです。もしカレントディレクトリが[MSDOS]であれば、

```
A>x：TOOL¥EDLIN ファイル名↵
```

で実行できます(ドライブ*x*：は、カレントドライブの場合省略できる)。

では、任意のカレントディレクトリから、ディレクトリ[TOOL]上の、エディタプログラム「EDLIN.

EXE」を使って、たった1行の文書ファイルを、さきほどと同様に、ディレクトリ[WP]上に作ってみましょう(図11.17)。なお、ここではいずれもカレントドライブ上で実習しています。

```
A>PATH; ↵ ……………パスの設定を解除する
A>PATH ↵ ……………確認する
パスは定義されていません.

A>CD ↵ ………カレントディレクトリを確認する
A:¥ ……………ルートディレクトリにいる

A>EDLIN WP¥TEST2 ↵ ……………パスが解除された状態で図11.16と同様にEDLINを起動しようとする
コマンドまたはファイル名が違います. ……………カレントディレクトリは[ルート]なので、[TOOL]上の
                                    プログラム「EDLIN.EXE」は実行できない
A>MSDOS¥TOOL¥EDLIN WP¥TEST2 ↵ ……………今回はパスを指定してEDLINを実行し、ディレクトリ[WP]上に
新しいファイルです. ……………EDLINが起動された        ファイル'TEST2」を作成する
*I ……………INSERTコマンドで、インサートモードにはいる
     1:*階層ディレクトリの実習　パスを解除して、WPにファイルTEST2を作ります。↵
     2:*^Z ↵ …………CTRL+Z でインサートモードを終わる
*E ……………ENDコマンドでEDLINを終了する                        文字を入力する

A>
```

図11.17 ディレクトリ[TOOL]上のエディタプログラム「EDLIN」をダイレクトに実行して、ファイルを作成する

```
A>DIR ¥WP ↵ ……………ファイル「TEST2」が作成されているディレクトリ[WP]を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥WP ……………表示されているディレクトリは[WP]

.            <DIR>      89-07-30   15:21
..           <DIR>      89-07-30   15:21
BUN          <DIR>      89-07-30   15:26
JXW          <DIR>      89-07-30   15:26
TEST1              72   89-07-30   16:58
TEST2              72   89-07-30   17:01 ……………確かに作成されている
        6 個のファイルがあります.
   756736 バイトが使用可能です.

A>TYPE ¥WP¥TEST2 ↵ ……………作成したファイルの内容をTYPEコマンドで確認する(TYPEコマン
                           ドは内蔵コマンドなので、パスの指定は必要ない)
階層ディレクトリの実習　パスを解除して、WPにファイルTEST2を作ります。

A>
```

図11.18 EDLINで作成された文書ファイルを確認する

図11.17の実行例でわかるように、作成されたファイルが書き込まれるディレクトリは、実行時のEDLINのコマンドで記述した、作成されるファイルのディレクトリを指定したパス名によります。そのパス名を省略した場合は、カレントディレクトリ上に作成されます。このことは前項でも述べましたが、すべてのコマンドやプログラムの実行についていえることですので注意してください。

次は、MDコマンドで子ディレクトリを作成した場合に、MS-DOSの内部で自動的に作成される2つのディレクトリ[.]と[..]の使い方 —— **ディレクトリの相対的な指定**についての実習です。

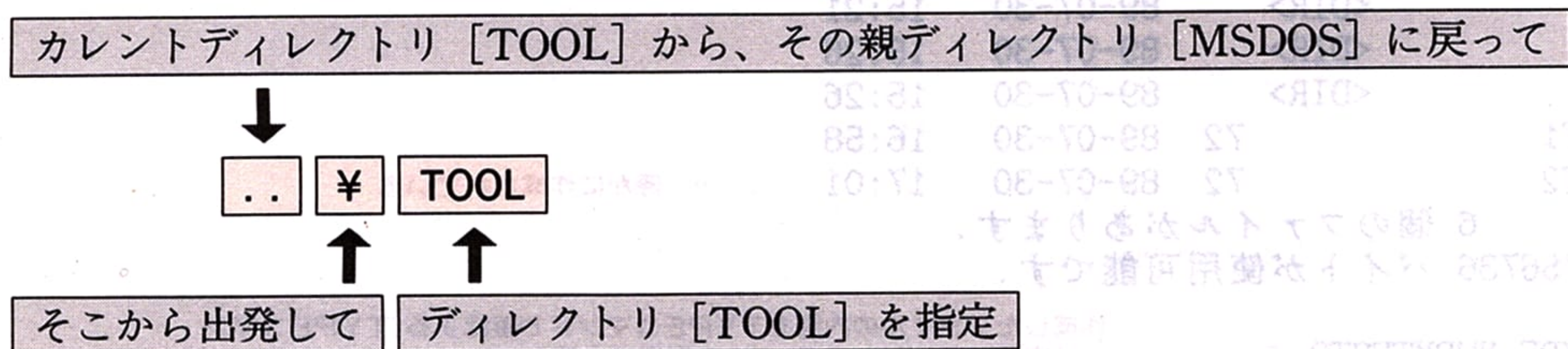
[.] ………カレントディレクトリを表す
[..] ………自分の親ディレクトリを表す

[.] については、これまでの実習で登場していますので、ここでは省略しましょう。[..] の使い方を、**図11.19**に示します。

```
A>CD ¥MSDOS ……………カレントディレクトリを[MSDOS]に移す

A>TYPE TEST2 ……………ファイル「TEST2」の内容を確認する
ファイルが見つかりません. ……………ファイル「TEST2」はこのディレクトリにはない

A>TYPE ..¥WP¥TEST2 ……今回は、[MSDOS]→[親ディレクトリ(ルート)]→[WP]のように
                        パスを指定して、ファイル「TEST2」の内容を確認する
階層ディレクトリの実習　パスを解除して、WPにファイルTEST2を作ります。…………表示された

A>
```

図11.19　自分の親ディレクトリを表す[..]による操作例1

図11.19で使用したパス名「..¥TOOL」は、次のように考えます。

カレントディレクトリ［TOOL］から、その親ディレクトリ［MSDOS］に戻って
↓
.. ¥ TOOL
↑ ↑
そこから出発して　ディレクトリ［TOOL］を指定

[..] を使った実行例を、もう一例示します(**図11.20**)。

```
A>CD ¥MSDOS¥TOOL ⏎ ……………カレントディレクトリを[TOOL]に移す

A>TYPE TEST1 ⏎ ……………ファイル「TEST1」の内容を確めようとする
ファイルが見つかりません．……………ファイル「TEST1」は、ここにはない

A>TYPE ¥WP¥TEST1 ⏎ ……………パスを[ルート]→[WP]のように指定して、ファイル「TEST1」の内容を確める
階層ディレクトリの実習　パスを設定して、WPにファイルTEST1を作ります。…………表示された

A>TYPE ..¥..¥WP¥TEST1 ⏎ ……今回はパスを[TOOL]→[親ディレクトリ(MSDOS)]→[その親ディレクトリ(ルート)]→[WP]のように指定して、ファイル[TEST1]の内容を確める
階層ディレクトリの実習　パスを設定して、WPにファイルTEST1を作ります。…………表示された

A>
```

図 11.20　自分の親ディレクトリを表す[..]による操作例 2

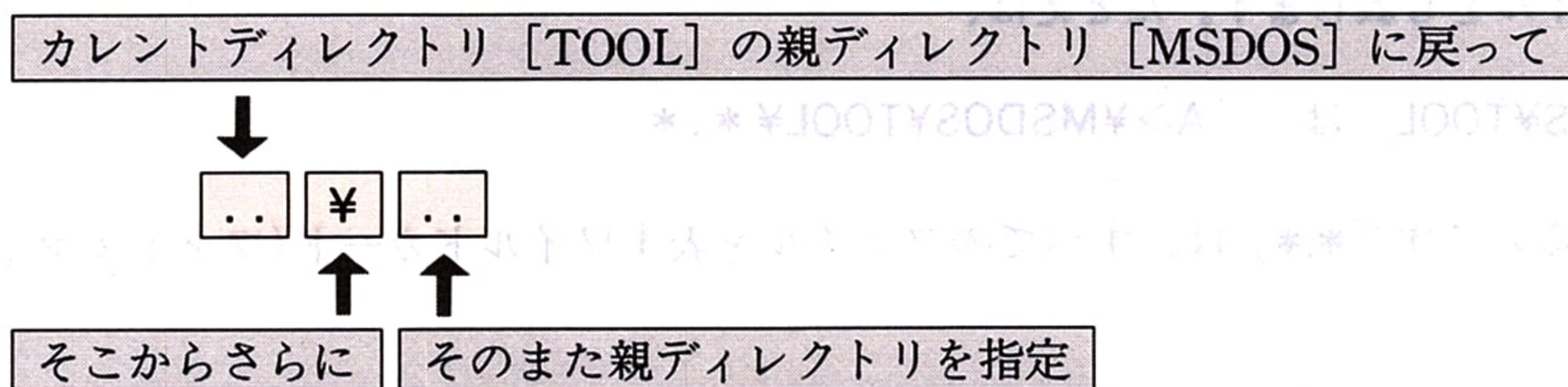
この場合の「..¥..」は、次のように考えます。

カレントディレクトリ［TOOL］の親ディレクトリ［MSDOS］に戻って

↓

.. ¥ ..

↑ ↑

そこからさらに　そのまた親ディレクトリを指定

それは結局、ルートディレクトリにあたるわけです。

さて本章では、階層ディレクトリをどのように利用するか、その基本的な運用法を実習しました。本章で解説したことを要約すれば、「10章以前で取り扱ってきたファイル名の拡張」ということになります。つまり、階層ディレクトリを使用せず、ルートディレクトリ上でのみ運用していた場合のファイル名は、

x：ABCD.XYZ

で表され、「ドライブ名＋ファイル名」によって、任意のファイルを完全に指定することができました。しかし、階層ディレクトリを利用する場合は、このような従来のファイル名に、さらにパス名を付加しなければなりません。つまり、

x：〈パス名〉ABCD.XYZ

で表され、このように「ドライブ名＋パス名＋ファイル名」によって、はじめて任意のファイルが完全に指定できるのです(ここでの〈パス名〉とは、たとえば「¥MSDOS¥TOOL¥」のようなパスを象

徴的に表している)。このように、階層ディレクトリを運用している場合のファイル名は、従来のファイル名に、そのパスを指定するためのパス名を付加したものと考えれば、今までどおりの取り扱いと変わるところはありません。

これらを整理して、**表 11.1** に示します。

階層ディレクトリを利用する／しない	完全なファイル名
階層ディレクトリを利用せず、ルートディレクトリ上のみで運用する場合	x:ABCD.XYZ (ドライブ名＋ファイル名)
階層ディレクトリを利用する場合	x:パス￥ABCD.XYZ (ドライブ名＋パス名＋ファイル名)

表 11.1　階層ディレクトリを利用する場合／しない場合の完全なファイル名の形式

また、パス名は、それによって指定されたディレクトリを表しますが、同時に、そのディレクトリ内のすべてのファイルをも表します。たとえば、

A＞￥MSDOS￥TOOL　は、　A＞￥MSDOS￥TOOL￥＊.＊

という意味でもあるのです(「*.*」は、すべてのファイルを表すワイルドカード(ファイルマッチ)記号)。したがって、

```
A>DEL ￥MSDOS￥TOOL ⏎
```

を実行すれば、ディレクトリ［TOOL］上に直接存在するすべてのファイルを削除することができます([TOOL］上に子ディレクトリが存在していても、それらのディレクトリ上のファイルにはいっさい影響しない)。

また、

A＞COPY　￥MSDOS￥TOOL　￥WP　は、A＞COPY　￥MSDOS￥TOOL￥＊.＊　￥WP

と同じ意味のコマンドです。つまり、ディレクトリ［TOOL］上に直接存在するすべてのファイルを、ディレクトリ［WP］上にコピーすることができるわけです。

では、これらの使い方に関する実行例を 2、3 示しておきましょう(**図 11.21**)。

```
A>COPY ¥MP ¥WP¥JXW ⏎
A:¥MP¥URIAGE.MP
A:¥MP¥MACHINE.MP
A:¥MP¥MITUMORI.MP
        3 個のファイルをコピーしました.

A>DIR ¥WP¥JXW ⏎

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥WP¥JXW

.             <DIR>      89-07-30   15:26
..            <DIR>      89-07-30   15:26
URIAGE   MP     55783    88-10-25   15:16
MACHINE  MP      7278    89-05-31   19:37
MITUMORI MP     51592    89-06-22   17:24
        5 個のファイルがあります.
   620544 バイトが使用可能です.

A>
```

ディレクトリ[MP]上のすべてのファイルを、ディレクトリ[JXW]上にコピーする。すべてのファイルを表すワイルドカード記号「*.*」が必要ないことに注目

DIRコマンドで結果を確認する

確かにコピーされている

同じことを実行するもう1つの方法

```
A>CD MP ⏎

A>COPY . ¥WP¥JXW ⏎
.¥URIAGE.MP
.¥MACHINE.MP
.¥MITUMORI.MP
        3 個のファイルをコピーしました.

A>
```

カレントディレクトリを[MP]に移す

自分のディレクトリ内(.)のすべてのファイルを、ディレクトリ[JXW]にコピーする。自分のディレクトリを示す「.」に注目

— 図 11.21 — (次ページに続く)

```
A>DEL ¥WP¥JXW ↵ ……………ディレクトリ[JXW]上のすべてのファイルを削除する
よろしいですか <Y/N>? Y ↵ ……………確認を求めてくる

A>DIR ¥WP¥JXW ↵ ……………DIRコマンドで結果を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥WP¥JXW

.            <DIR>      89-07-30   15:26
..           <DIR>      89-07-30   15:26  ←確かにすべてのファイルが削除されている
        2 個のファイルがあります.
  737280 バイトが使用可能です.

A>
```

図 11.21 ディレクトリの指定が、そのディレクトリに直接存在する全ファイルを表すことの実行例

最後は、ディレクトリを削除するRDコマンド(Remove Directory)の実習です。カレントディレクトリが［ルート］のとき、ディレクトリ［TOOL］の削除を試みます。なお現在、［TOOL］上には、エディタ「EDLIN」のプログラムファイルが存在しています。

```
A>RD MSDOS¥TOOL ↵ ……………ディレクトリ[MSDOS]上のサブディレクトリ[TOOL]を削除しようとする
パスの指定が違うかディレクトリでないか   ｝[TOOL]上にはファイル「EDLIN.EXE」が存在
またはディレクトリが空ではありません.   ｝するのでディレクトリの削除は拒否された

A>
```

図 11.22 ファイルが存在しているディレクトリの削除を試みる

このように、ファイルやディレクトリが収容されているディレクトリを削除することはできません。ディレクトリを削除するためには、なにも収容されていない空のディレクトリにしなければなりません。では、まず存在しているファイルを削除して、空のディレクトリにしてから、再度RDコマンドを実行してみましょう。

```
A>DIR ¥MSDOS¥TOOL ↵ ……………削除しようとするディレクトリ[TOOL]の内容を確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥MSDOS¥TOOL

.            <DIR>      89-08-03   15:05
..           <DIR>      89-08-03   15:05
EDLIN    EXE      7922  88-07-13    0:00 ……………ファイルが存在している
        3 個のファイルがあります.
   747520 バイトが使用可能です.

A>DEL MSDOS¥TOOL¥*.* ↵ ……………[TOOL]上のファイルをすべて削除する
よろしいですか <Y/N>? Y ↵ ……………「Y」を入力する

A>CD ¥MSDOS¥TOOL ↵     }
                        } 試みに、カレントディレクトリを[TOOL]に移して、
A>RD TOOL ↵            } 自分自身を削除してみる
パスの指定が違うかディレクトリでないか    }
またはディレクトリが空ではありません.      } カレントディレクトリは削除できない

A>CD ¥ ↵ ……………再びカレントディレクトリを[ルート]に戻す

A>RD MSDOS¥TOOL ↵ ……………サブディレクトリ[TOOL]を削除する

A>DIR MSDOS ↵ ……………確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥MSDOS

.            <DIR>      89-07-30   15:21
..           <DIR>      89-07-30   15:21
FORMAT   EXE     97766  88-07-13    0:00
SYS      EXE     25480  88-07-13    0:00
DISKCOPY EXE     19604  88-07-13    0:00
CHKDSK   EXE     10384  88-07-13    0:00
TREE     EXE      9608  88-07-13    0:00 ←─[MSDOS]上のサブディレクトリ
        7 個のファイルがあります.             [TOOL]は削除されている
   748544 バイトが使用可能です.

A>
```

図 11.23　空のディレクトリに対してディレクトリの削除を試みる

なお、カレントディレクトリが自分自身の[TOOL]以外であれば、どこであっても、

```
A>RD  x:¥MSDOS¥TOOL ↵
```

とすれば、目的のディレクトリ、[TOOL]を削除することができます(ドライブ名 x：は、カレントドライブの場合省略できる)。

■1枚のディスク、1台のハードディスクに収容可能なディレクトリ数について

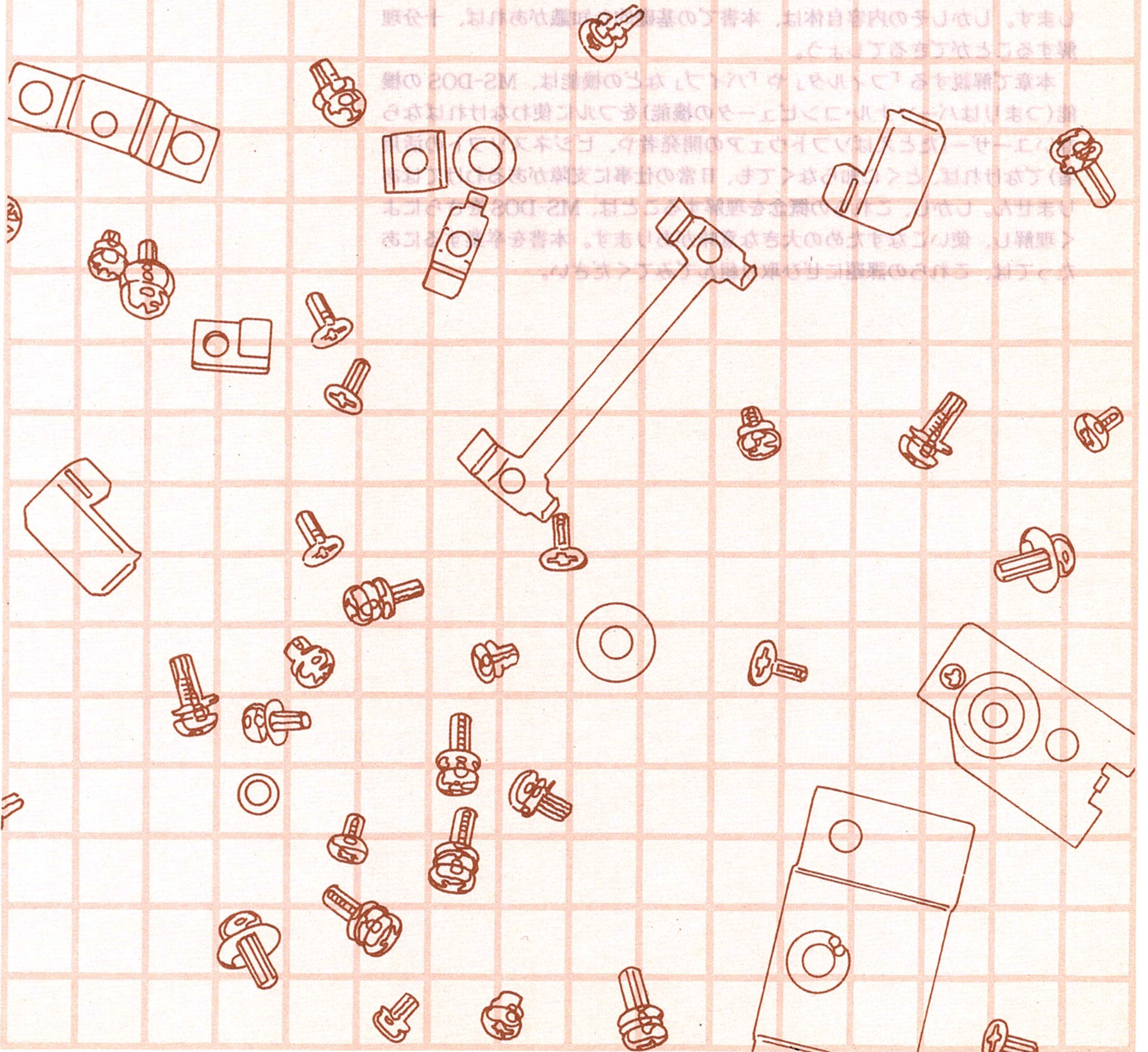
MS-DOSでは、1つのディスクに収容可能なファイルの数や、ディレクトリの数には、フロッピーディスクやハードディスクなどの種類に関係なく制限がありません。それぞれのディスクの容量がいっぱいになるまでは、自由にいくつでも作成可能です。ただし、ルートディレクトリ上に直接収容できるディレクトリの数(ファイルの数と子ディレクトリの数の合計)には、次の制限があります。

- 8インチ1Dの場合　68個
- 5インチ2DD、3.5インチ2DDの場合　112個
- 8インチ2D、5インチ2HD、3.5インチ2HDの場合　192個
- 20Mバイト標準フォーマットのハードディスクの場合　1504個
- 40Mバイト拡張フォーマットのハードディスクの場合　3042個

言い換えれば、階層ディレクトリを利用せずに、ルートディレクトリ上でのみ作業する場合は、上記の制限を受けることになります。

しかし、階層ディレクトリを利用する場合の子ディレクトリには、上記のような数の制限はなくなるわけです。極端な例では、ルートディレクトリに1つの子ディレクトリを作り、その子ディレクトリ上で作業すれば、ファイルやディレクトリの数の制限はありません。ディスクの容量の許す範囲で、無限にファイルやディレクトリを作ることができます。

なお、ディレクトリを作成すると、ディスクのメモリが消費されます。フロッピーディスクの場合は、1つのディレクトリを作成するだけで(そこにファイルを収容しなくても)、ディスクの使用可能容量が1Kバイト消費されます(8インチ2D、5インチ2HD、3.5インチ2HDの場合)。また、上記の20Mバイト標準フォーマットのハードディスクの場合は、1つのディレクトリにつき8Kバイト、40Mバイト拡張フォーマットの、従来のインターフェイス(SASIインターフェイス)で接続されている場合には16Kバイト、SCSIインターフェイスで接続されている場合には4Kバイトが消費されます(それぞれのディスクの「クラスタサイズ」分にあたる)。ディレクトリも「ただ」では作れないことをいちおう頭に入れておいてください。

12章
フィルタとパイプおよびリダイレクト

MS-DOS は、ミニコンピュータや、高性能ワークステーションの有力OS であるUNIX を指向しています。MS-DOS に採り入れられている「階層ディレクトリ」(11 章)、「デバイスファイル」(9.1 節)、「リダイレクト」(9.3 節、本章)、「フィルタ」や「パイプ」(本章)などは、UNIX 流の概念なのです。

これらはけっしてむずかしい考え方ではありませんが、MS-DOS によって初めてOS というものに接する人にとっては、少々難解かも知れません。これらの概念は、比較的エレガントなものであり、OS 全体の知識を必要とします。しかしその内容自体は、本書での基礎的な知識があれば、十分理解することができるでしょう。

本章で解説する「フィルタ」や「パイプ」などの機能は、MS-DOS の機能(つまりはパーソナル・コンピュータの機能)をフルに使わなければならないユーザー(たとえばソフトウェアの開発者や、ビジネスソフトの活用者)でなければ、とくに知らなくても、日常の仕事に支障があるわけではありません。しかし、これらの概念を理解することは、MS-DOS をさらによく理解し、使いこなすための大きな意味があります。本書を卒業するにあたっては、これらの課題にぜひ取り組んでみてください。

12.1 フィルタ

MS-DOSでいう**フィルタ**とは、エアコンに付いている空気中のゴミを取るエアフィルタとか、カメラのレンズに取り付けるスカイライトフィルタとか、オーディオアンプのローパスフィルタなどの「フィルタ」と同じ意味のものです。つまり、入力したものから、特定なものを取り除いたり、取り出したりする —— 入力したデータに対して特定の処理を加えて出力する、ソフトウェア的なフィルタのことを指します。

MS-DOSのフィルタは、それぞれ次に示すコマンドを利用して機能します。フィルタとして利用できるコマンドには、次の3つがあります。

FIND ………任意の文字列をサーチ(検索)する機能
SORT ………文字列データをソート(並べ換え)する機能
MORE ………ディスプレイ出力を1画面ごとに表示する機能

これらは、いわゆる「フィルタ」ですから、それぞれに「材料」(データ)を入力してやらなければなりませんし、フィルタを通ったあとの「加工された材料」の行き先(出力先)を指定しておかなければなりません。フィルタに入力する材料に当るものには、次の2つがあります。

- ファイル(ディスク上のファイル、およびデバイスファイル)
- 各種のコマンドを実行した際の結果の出力

そして「何を」「どのフィルタへ」入力するか、そこで処理したものを「どこへ」出力するかを指定するのが次節で解説する**パイプ**であり、9.3節でその一部を紹介した**リダイレクト**なのです(図12.1)。

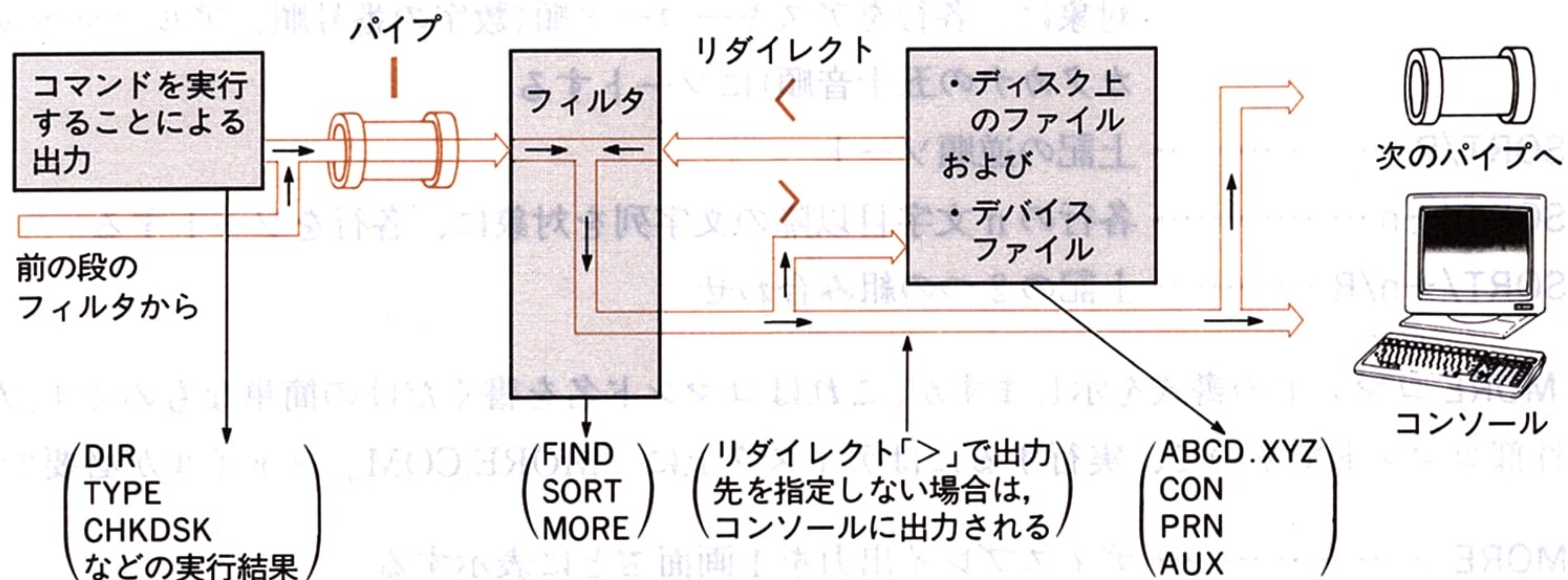



図12.1 フィルタ、パイプおよびリダイレクトの概念図

フィルタへの入力元の指定は必ず行わなければなりませんが、出力先の指定は省略することができます。フィルタの場合に限らず、出力先を省略した場合は、自動的にコンソールCON(通常はディスプレイ)が指定されますので、出力はディスプレイに表示されることになります。

たとえば、DIR コマンドを例にとると、

```
A>DIR↵
```

の意味は本来、

```
A>DIR > CON↵
```

ということであり、「> CON」の部分を省略したものと考えることができます。

さて、3つのフィルタコマンドの1つ、**FIND** コマンドは、7.13節においてはフィルタとしてではなく、1つのコマンドとして実行例を示しましたが、ここではフィルタコマンドとしての使い方を実習します。その場合のコマンドの書式を次に示します。なお、このコマンドは外部コマンドですので、実行するにはディスク上にプログラムファイル、「FIND.EXE」が必要です。

FIND ''*xxx*'' ……… 入力されたものから、任意の文字列「*xxx*」を捜し出し、それが含まれるすべての行を出力する

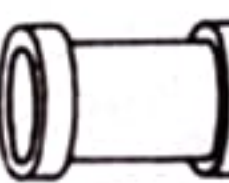
FIND/V ''*xxx*'' …… 入力されたものから、任意の文字列「*xxx*」を捜し出し、それが含まれないすべての行を出力する

次に SORT コマンドの書式を示します。このコマンドも外部コマンドですので、実行するにはディスク上にプログラムファイル「SORT.EXE」が必要です。

SORT …………………… 入力されたデータの各行の1文字目以降(つまり先頭から)の文字列を対象に、各行をアスキーコード順(数字の番号順、アルファベット順、カタカナの五十音順)にソートする

SORT/R ……………… 上記の逆順ソート

SORT/+n …………… 各行のn文字目以降の文字列を対象に、各行をソートする

SORT/+n/R ………… 上記の2つの組み合わせ

次に MORE コマンドの書式を示しますが、これはコマンド名を書くだけの簡単なものです。ただし、これも外部コマンドですので、実行するにはディスク上に「MORE.COM」ファイルが必要です。

MORE …………………… ディスプレイ出力を1画面ごとに表示する

以上の3つのフィルタに関するコマンドの実習は、次節で行いましょう。

12.2 パイプおよびリダイレクト

12.2.1 パイプ

「パイプ」とは、水道管やガス管、石油のパイプラインなどの「パイプ」の意味であり、ある所からある所へ、ものを送り込む機能を意味します。つまり、あるコマンドの実行により出力されるデータを、次のコマンドに入力する機能を意味し、このパイプをコマンドで記述するには、記号「|」で表します(「」の意味と考えればよい)。この記号は、JIS キーボード上では、多くのものが ¥ キーをシフトして入力するようになっています。また、コンピュータの機種やプリンタの機種によっては、「¦」と表示される場合もあります。

ではパイプの機能を使った基本的な実行例をいくつか示しましょう。最初は、最も代表的なもので、多くのマニュアル類に必ず出てくる DIR コマンドを使ったものです。パイプ記号「|」の前後には、スペースを入れても入れなくてもかまいません。パイプのコマンドを実行する際に注意することは、MS-DOS がパイプ処理を行うための内部的な作業用に、カレントドライブが使われることです(読み/書きが行われる)。したがって、カレントドライブに当るディスクが書き込み禁止になっていたり、残り容量が不足している場合は、パイプは実行できません。このことに注意してください。

では次のコマンドを実行します。なお、各コマンドの実行例には、MS-DOS の常用システムディスクを使い、すべてをカレントドライブのカレントディレクトリで行うことにします。

```
A>DIR | MORE↵
```

このコマンドは、DIR コマンドによる表示出力を、MORE フィルタに入力してその処理を行わせ、

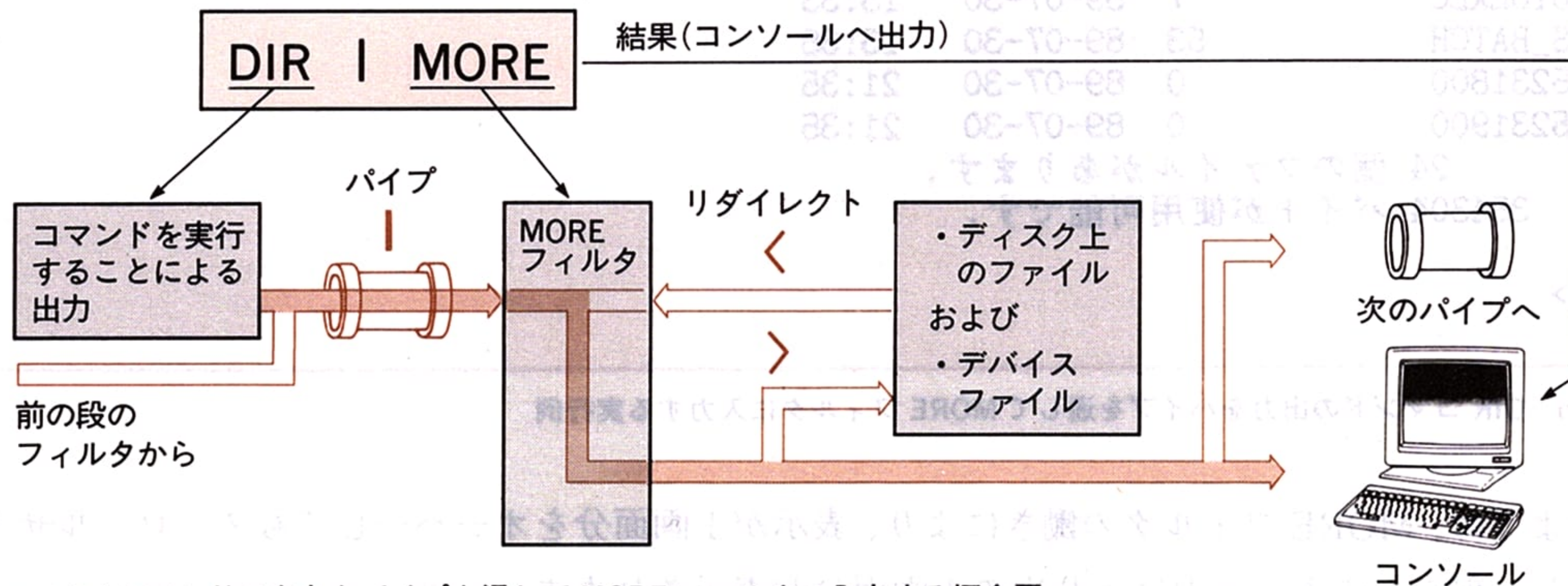



図 12.2a DIR コマンドの出力をパイプを通して MORE フィルタに入力する概念図

その結果をコンソールに表示するものです(本来は「DIR | MORE > CON」であるが、出力先の表示は省略されている)(図 12.2a、図 12.2b)。

パイプ

```
A>DIR | MORE ↵
 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

COMMAND  COM    24931  88-07-13    0:00
CONFIG   SYS       87  89-07-29   17:25
ATOK6A   SYS    54271  87-06-15   12:00
ATOK6B   SYS    21372  87-06-15   12:00
ATOK     DIC   454144  89-07-30   16:50
PRINT    SYS     5855  88-07-13    0:00
RSDRV    SYS     7352  88-07-13    0:00
MOUSE    SYS     3985  88-07-13    0:00
FORMAT   EXE    97766  88-07-13    0:00
DISKCOPY EXE    19604  88-07-13    0:00
CHKDSK   EXE    10384  88-07-13    0:00
SYS      EXE    25480  88-07-13    0:00
FC       EXE    15302  88-07-13    0:00
FIND     EXE     6573  88-07-13    0:00
MORE     COM      303  88-07-13    0:00
SORT     EXE     2138  88-07-13    0:00
TREE     EXE     9608  88-07-13    0:00
ATTRIB   EXE     9154  88-07-13    0:00
EDLIN    EXE     7922  88-07-13    0:00
-- More -- x
```

カレントドライブ(A:)に対してDIRコマンドを実行し、その表示出力をMOREフィルタに入力する。その出力はコンソールに表示される。(もしMORE.COMがドライブB：上のカレントディレクトリに存在する場合は、「B：MORE」とする)

1画面分が表示されてスクロールがいったんストップする

適当なキーを入力することにより、次の1画面が表示される

```
README   DOC    10578  88-07-13    0:00
AUTOEXEC            7  89-07-30   13:33
TS_BATCH           53  89-07-30   13:38
15231800            0  89-07-30   21:35
15231900            0  89-07-30   21:35
         24 個のファイルがあります.
     354304 バイトが使用可能です.

A>
```

続いて残りの部分が表示される

図 12.2b　DIR コマンドの出力をパイプを通して MORE フィルタに入力する実行例

このように、MORE フィルタの働きにより、表示が 1 画面分をオーバーしてもスクロールせずにストップし、何らかのキー入力により次の 1 画面分が表示されます。

次の実行例は、同じ DIR コマンドによる出力を、今度は SORT フィルタに入力し、表示されるファイルのデータを、ファイル名のアルファベット順にソートするものです(図 12.3a、図 12.3b)。次のコマンドを実行します。

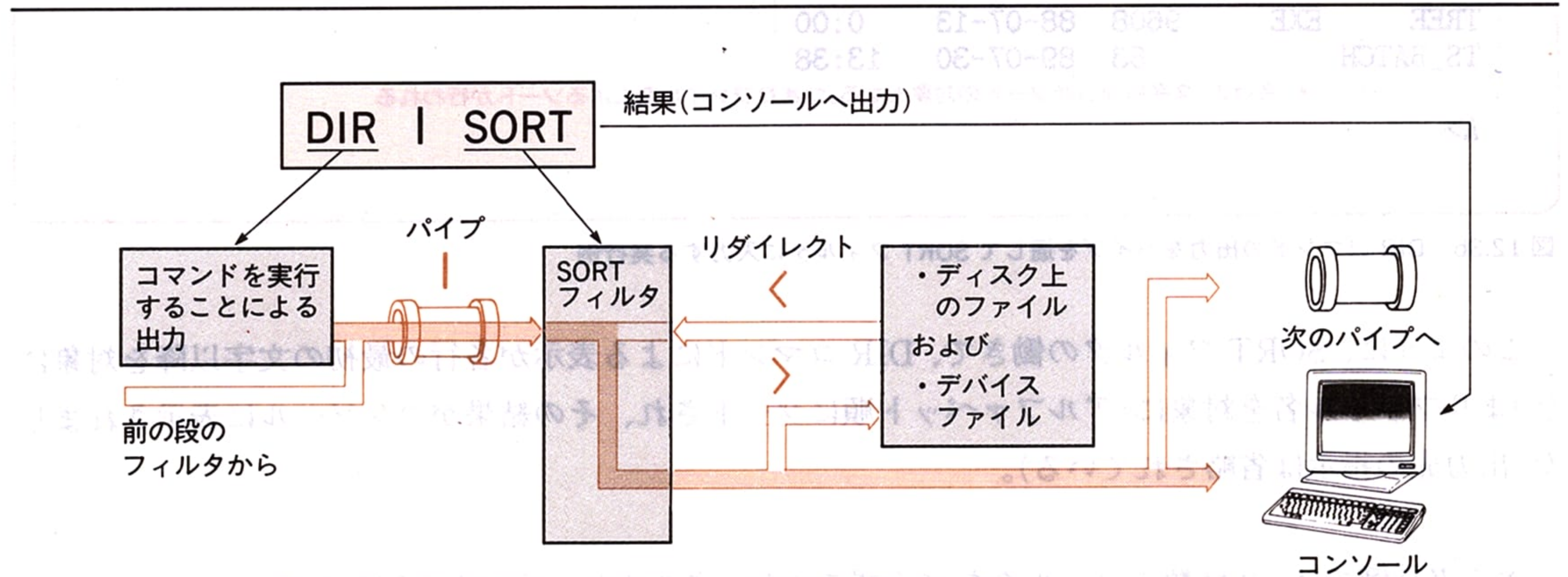
```
A>DIR | SORT ⏎
```


図 12.3a　DIR コマンドの出力をパイプを通して SORT フィルタに入力する概念図


```
A>DIR | SORT ⏎
       24 個のファイルがあります.
   354304 バイトが使用可能です.
 ディレクトリは A:¥
 ドライブ A: のディスクのボリュームラベルはありません
15260E00                0  89-07-30  21:38
15260F00                0  89-07-30  21:38
ATOK      DIC      454144  89-07-30  16:50
ATOK6A    SYS       54271  87-06-15  12:00
ATOK6B    SYS       21372  87-06-15  12:00
ATTRIB    EXE        9154  88-07-13   0:00
AUTOEXEC                7  89-07-30  13:33
CHKDSK    EXE       10384  88-07-13   0:00
COMMAND   COM       24931  88-07-13   0:00
CONFIG    SYS          87  89-07-29  17:25
DISKCOPY  EXE       19604  88-07-13   0:00
EDLIN     EXE        7922  88-07-13   0:00
FC        EXE       15302  88-07-13   0:00
FIND      EXE        6573  88-07-13   0:00
```

— 図 12.3b —(次ページに続く)

```
FORMAT   EXE    97766  88-07-13    0:00
MORE     COM      303  88-07-13    0:00
MOUSE    SYS     3985  88-07-13    0:00
PRINT    SYS     5855  88-07-13    0:00
README   DOC    10578  88-07-13    0:00
RSDRV    SYS     7352  88-07-13    0:00
SORT     EXE     2138  88-07-13    0:00
SYS      EXE    25480  88-07-13    0:00
TREE     EXE     9608  88-07-13    0:00
TS_BATCH           53  89-07-30   13:38
A>
```

各行の1文字目からがソートの対象となる。つまりファイル名によるソートが行われる

図 12.3b　DIR コマンドの出力をパイプを通して SORT フィルタに入力する実行例

このように、SORT フィルタの働きで、DIR コマンドによる表示が各行の最初の文字以降を対象に(つまりファイル名を対象に)アルファベット順にソートされ、その結果がコンソールに表示されました(出力先の指示は省略されている)。

次の実行例では、2種類のフィルタをパイプでつないでみましょう。上と同じ DIR コマンドによる出力を、FIND フィルタに入力し、そこで処理された出力をそのまま SORT フィルタへ入力してその結果を表示するものです。次のコマンドを実行します。

```
A>DIR | FIND "SYS" | SORT ⏎
```

このコマンドは、まず DIR コマンドによる出力を、FIND フィルタに入力して、文字列「SYS」が

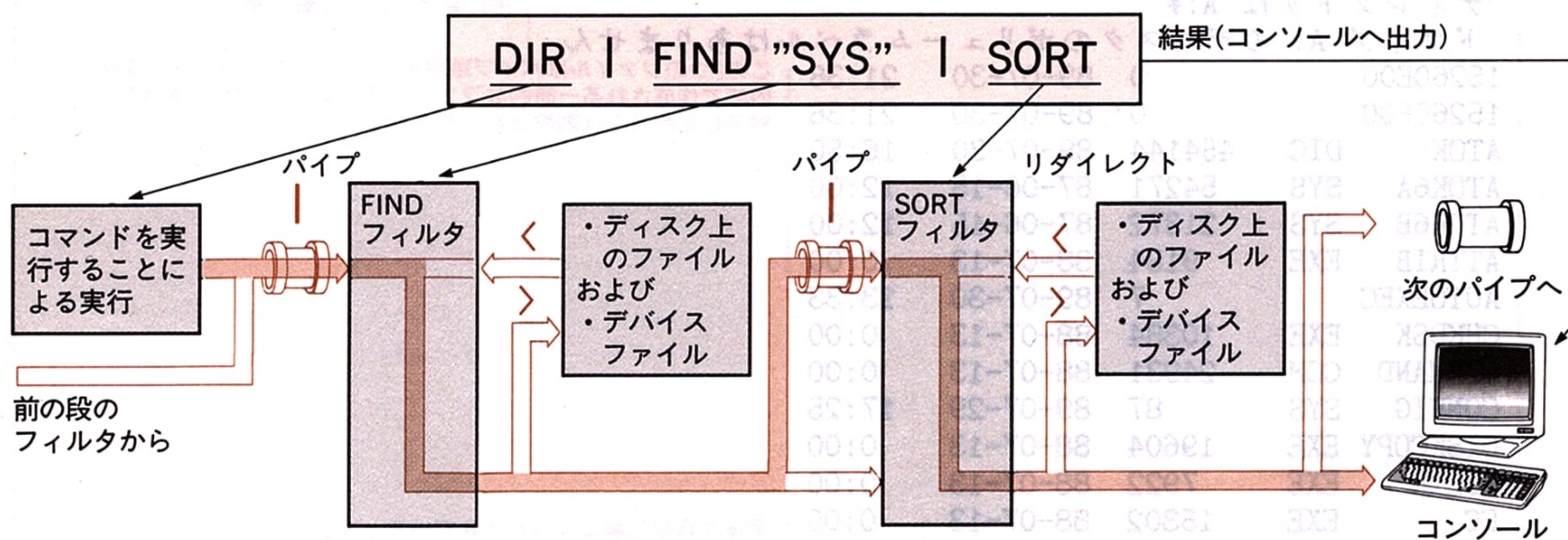



図 12.4a　DIR コマンドの出力をパイプを通して FIND フィルタと SORT フィルタに入力する概念図

含まれる行のみを捜し出します。次に、その結果をSORTフィルタへ入力してABC順にソートし、結果をコンソールに表示するものです(出力先の指示は省略されている)(図12.4a、図12.4b)。

```
     ┌─パイプ      ┌─パイプ
A>DIR | FIND "SYS" | SORT ⏎
ATOK6A   SYS    54271  87-06-15   12:00
ATOK6B   SYS    21372  87-06-15   12:00
CONFIG   SYS       87  89-07-29   17:25
MOUSE    SYS     3985  88-07-13    0:00
PRINT    SYS     5855  88-07-13    0:00
RSDRV    SYS     7352  88-07-13    0:00
SYS      EXE    25480  88-07-13    0:00

A>
```


図12.4b　DIRコマンドの出力をパイプを通してFINDフィルタとSORTフィルタに入力する実行例

このように、ファイル名のファイルタイプが「SYS」である行が捜し出され、さらにアルファベット順にソートされています(図12.2を参照)。

もし、結果の表示データが多く、1画面に収まらない場合には、さきほどのコマンドラインの後ろに、さらにパイプを使ってMOREフィルタを通し、

```
A>DIR | FIND "SYS" | SORT | MORE ⏎
```

のようにすればよいわけです。

また、この結果を画面ではなく、プリンタに出力したい場合は、次節で解説するリダイレクトを使って、出力先の指定を省略せずに、

```
A>DIR | FIND "SYS" | SORT > PRN ⏎
```

のように、「> PRN」を付け加えます。

また同様に、この結果をディスク上のファイルにすることもできます。上記コマンドの「PRN」を、任意のファイル名(たとえば、「ABCD.XYZ」)に変更すれば、そのファイル名でファイルが作成され、上の実行結果がディスクにセーブされます。各自で試してみてください。

以上は、パイプやフィルタの基本的な使い方の例ですが、さらにこれらの応用例を、本章の最後に示します。

12.2.2 リダイレクト

リダイレクトは、記号「>」と「<」によって機能し、それぞれは次のような意味があります。

> ……あるコマンドの実行によって出力されるデータを、ファイルに出力する(つまりディスクファイルにセーブしたり、デバイスファイルに出力する)

< ……ファイル(ディスクファイルやデバイスファイル)の内容を、あるコマンドに入力する

ではリダイレクトの基本的な実行例を示しましょう。DIR コマンドの実行による表示出力をファイルに出力するコマンド、

```
A>DIR  >  ファイル名↵
```

の実行例は、9.3 節で行いましたので、もう一度確認しておいてください。

ここではまず、前項で解説したパイプと組み合わせて、

```
A>DIR  |  FIND  "SYS"  >  ABCD.XYZ↵
```

というコマンドを実行してみましょう。このコマンドは、DIR コマンドの表示出力から、文字列「SYS」が含まれるすべての行を捜し出し、その出力を「ABCD.XYZ」という名前のファイルにセーブします(そのファイルが新しく作成される)(図 12.5a、図 12.5b)。リダイレクト記号「>」や「<」は、パイプ記号「|」と同様に、その前後にスペースを入れても入れなくても、どちらでもかまいません。なお、実行例はすべてカレントドライブのカレントディレクトリ上で行っています。

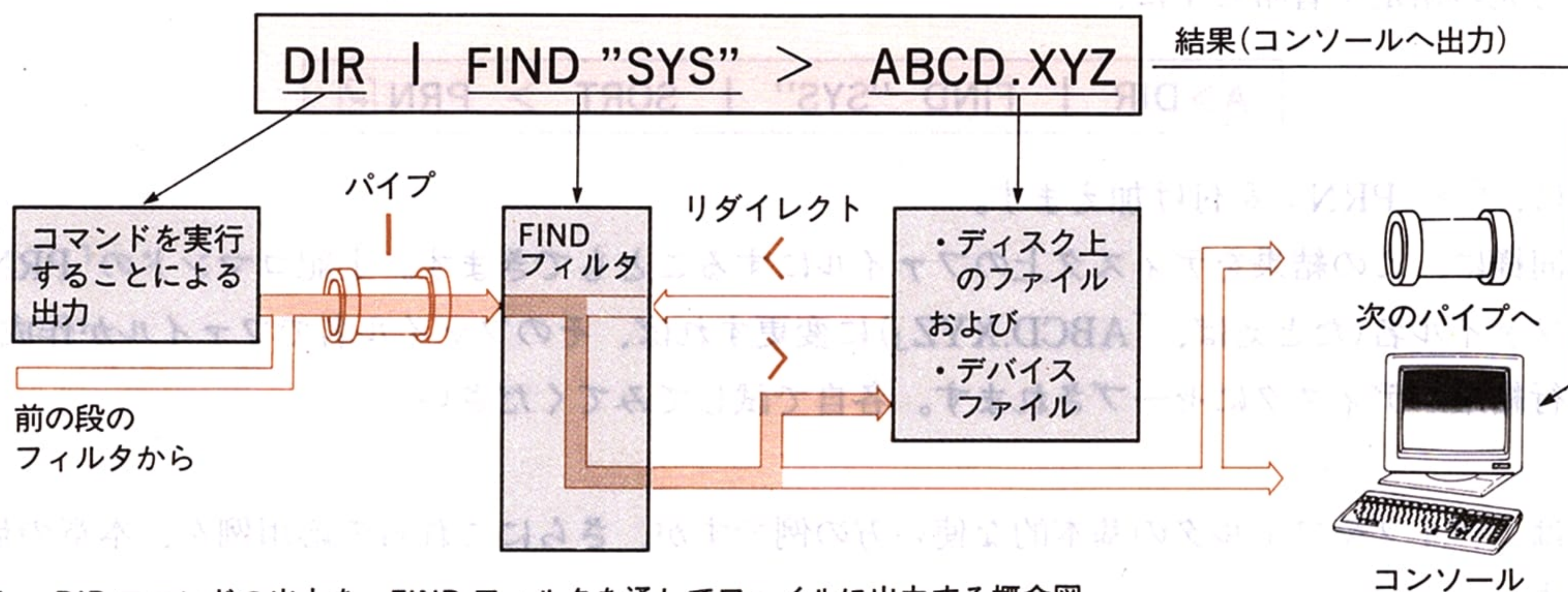
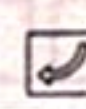



図 12.5a　DIR コマンドの出力を、FIND フィルタを通してファイルに出力する概念図

```
        ┌─ パイプ      ┌─ リダイレクト
A>DIR | FIND "SYS" > ABCD.XYZ ↵ ……DIRコマンドを実行し、その表示出力をFINDフィルタ
                                  に入力して文字列「SYS」を捜し、それを含む行を、ファ
                                  イル「ABCD.XYZ」を新たに作成して格納する

A>DIR ABCD.XYZ ↵ ……作成されたファイルを確認する

 ドライブ A: のディスクのボリュームラベルはありません.
 ディレクトリは A:¥

ABCD     XYZ       287  89-07-30   21:40
        1 個のファイルがあります.
  355328 バイトが使用可能です.

A>TYPE ABCD.XYZ ↵ ……作成されたファイルの内容をTYPEコマンドで確認する
CONFIG   SYS        87  89-07-29   17:25
ATOK6A   SYS     54271  87-06-15   12:00
ATOK6B   SYS     21372  87-06-15   12:00
PRINT    SYS      5855  88-07-13    0:00   内容は文字列「SYS」が含まれるすべての行である
RSDRV    SYS      7352  88-07-13    0:00
MOUSE    SYS      3985  88-07-13    0:00
SYS      EXE     25480  88-07-13    0:00

A>
```

図 12.5b　DIR コマンドの出力を、FIND フィルタを通してファイルにセーブする実行例

このように、DIR コマンドの表示出力から、ファイル名に「SYS」を含む行が捜し出され、その結果が新たに作成された ABCD.XYZ というファイルにセーブされています。またこのファイル名、「ABCD.XYZ」を「PRN」に変更すれば、ファイル「ABCD.XYZ」にセーブされた内容と同じものが、そのままプリンタに出力されます。つまり、デバイスファイルに出力されるわけです。プリンタへの出力の実習は、すでに 9.3 節で行っていますので、そちらを参照してください。

また同様に、コンソールに出力する場合は、「ABCD.XYZ」を「CON」に変更すればよいわけですが、すでに述べたように、「>　ABCD.XYZ」の部分を記述せず、出力先の指定を省略すれば、自動的にコンソール(CON)に出力されます。

さて次は「<」を使い、図 12.5の実行例で作成されたファイル、「ABCD.XYZ」を、SORT フィルタに入力する実行例です。次のコマンドを実行します。

```
A>SORT/+13  <  ABCD.XYZ ↵
```

このコマンドは、ファイル「ABCD.XYZ」を SORT フィルタに入力して、各行(DIR コマンドによる表示の各行)の 13 文字目以降の文字列を対象に(そこは、それぞれのファイルの大きさを示すバイト数の項目の先頭にあたる)、各行をソートして、その結果をコンソールに表示します(出力の指示が省略されているため「CON」に出力される)(図 12.6a、図 12.6b)。

なお、DIR コマンドによる表示出力の各行の構成と文字数との関係は次のようになっています。

ファイル名　ファイル容量(バイト数)　作成年月日　作成時刻

COMMAND □□ COM □□□□□ 24931 □□ 88-07-13 □□□ 0：00

文字数番号　1 2 3 4 5………　13………　24………　34………

SORT / +13 < ABCD.XYZ
結果(コンソールへ出力)
パイプ
リダイレクト
コマンドを実行することによる出力
前の段のフィルタから
SORT フィルタ
・ディスク上のファイル
および
・デバイスファイル
次のパイプへ
コンソール

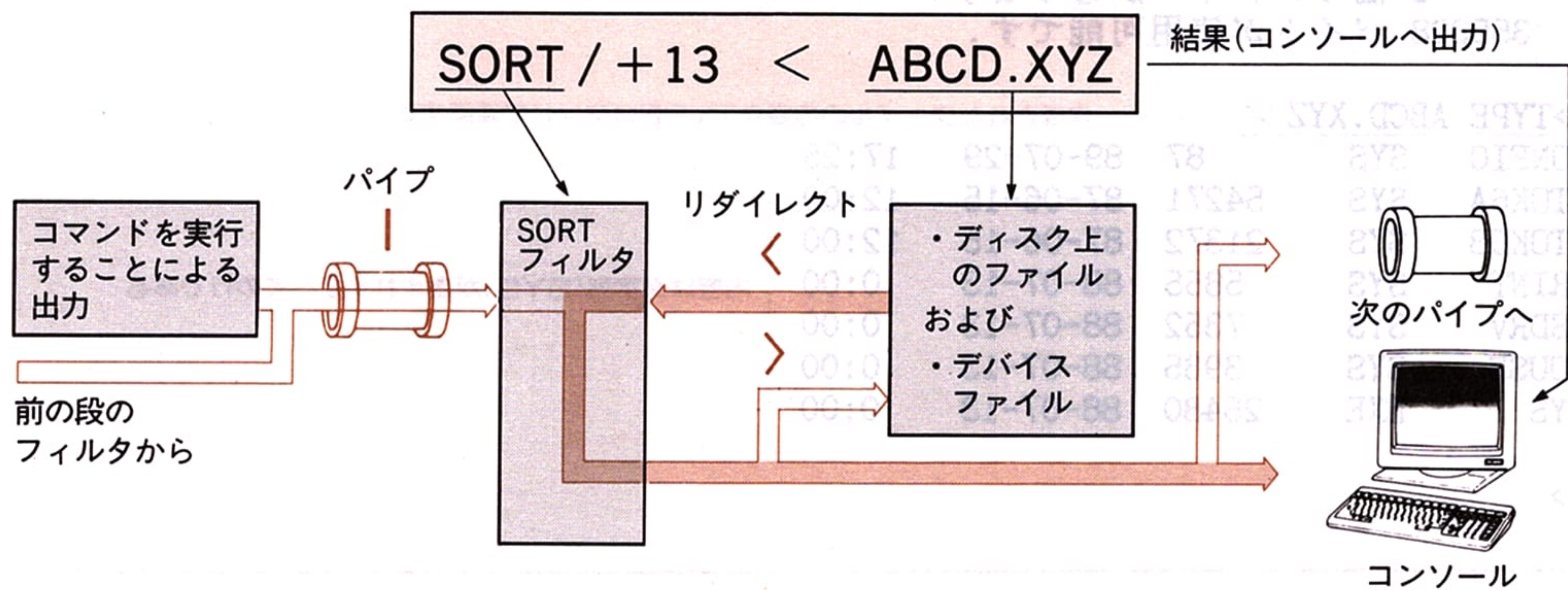



図 12.6a　ファイル ABCD.XYZ を SORT フィルタに入力して、コンソールに出力する概念図

```
A>SORT/+13 < ABCD.XYZ
CONFIG   SYS        87  89-07-29  17:25
MOUSE    SYS      3985  88-07-13   0:00
PRINT    SYS      5855  88-07-13   0:00
RSDRV    SYS      7352  88-07-13   0:00
ATOK6B   SYS     21372  87-06-15  12:00
SYS      EXE     25480  88-07-13   0:00
ATOK6A   SYS     54271  87-06-15  12:00
A>
```

リダイレクト

図12.5で作成したファイルABCD.XYZの内容を、各行の13文字目からソートする

ファイルサイズの小さい順にソートされている

ファイルサイズ

SORT/+13により、ここから右の部分がソートの対象となる。
つまりファイルサイズによるソートが行われる

図 12.6b　ファイル ABCD.XYZ を SORT フィルタに入力して、コンソールに出力する実行例

次に「<」を使って、同じファイル「ABCD.XYZ」をFINDフィルタに入力する実行例を示しましょう。次のコマンドを実行します。

```
A>FIND "06-15" < ABCD.XYZ ⏎
```

このコマンドは、ファイル「ABCD.XYZ」から、文字列「06-15」(6月15日作成のもの)が含まれるすべての行を捜し出して、コンソールに表示します(出力先の指示が省略されているため)(**図12.7a、図12.7b**)。

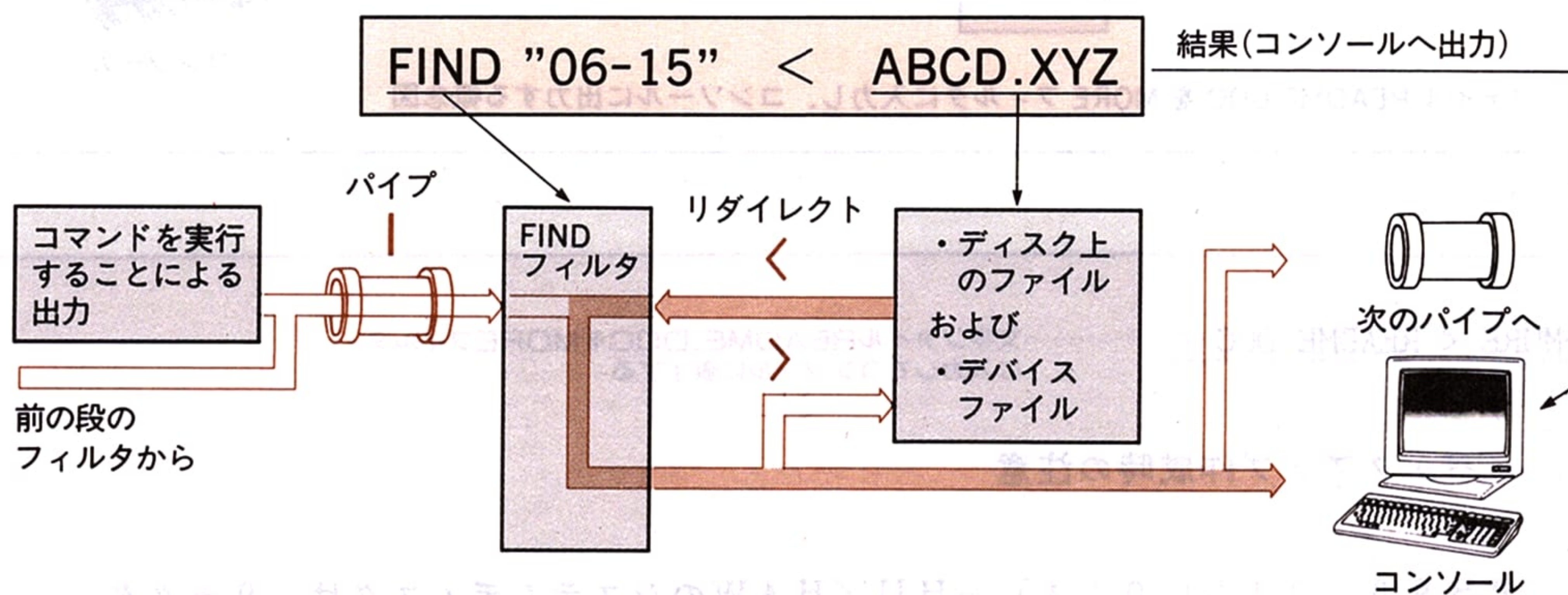



図 12.7a　ファイル ABCD.XYZ を FIND フィルタに入力し、コンソールに出力する概念図

```
A>FIND "06-15" < ABCD.XYZ ⏎
ATOK6A   SYS    54271  87-06-15   12:00
ATOK6B   SYS    21372  87-06-15   12:00

A>
```

図12.5で作成したファイルABCD.XYZの内容から、文字列「06-15」が含まれる行のみ取り出す

6月15日作成のファイルはこれだけあった

図 12.7b　ファイル ABCD.XYZ を FIND フィルタに入力し、コンソールに出力する実行例

もう1つ「<」を使った実行例を示します(**図12.8a、図12.8b**)。

```
A>MORE < README.DOC ⏎
```

このコマンドは、文書ファイル「README.DOC」をMOREフィルタに入力し、1画面ごとに区切ってコンソールに表示します。

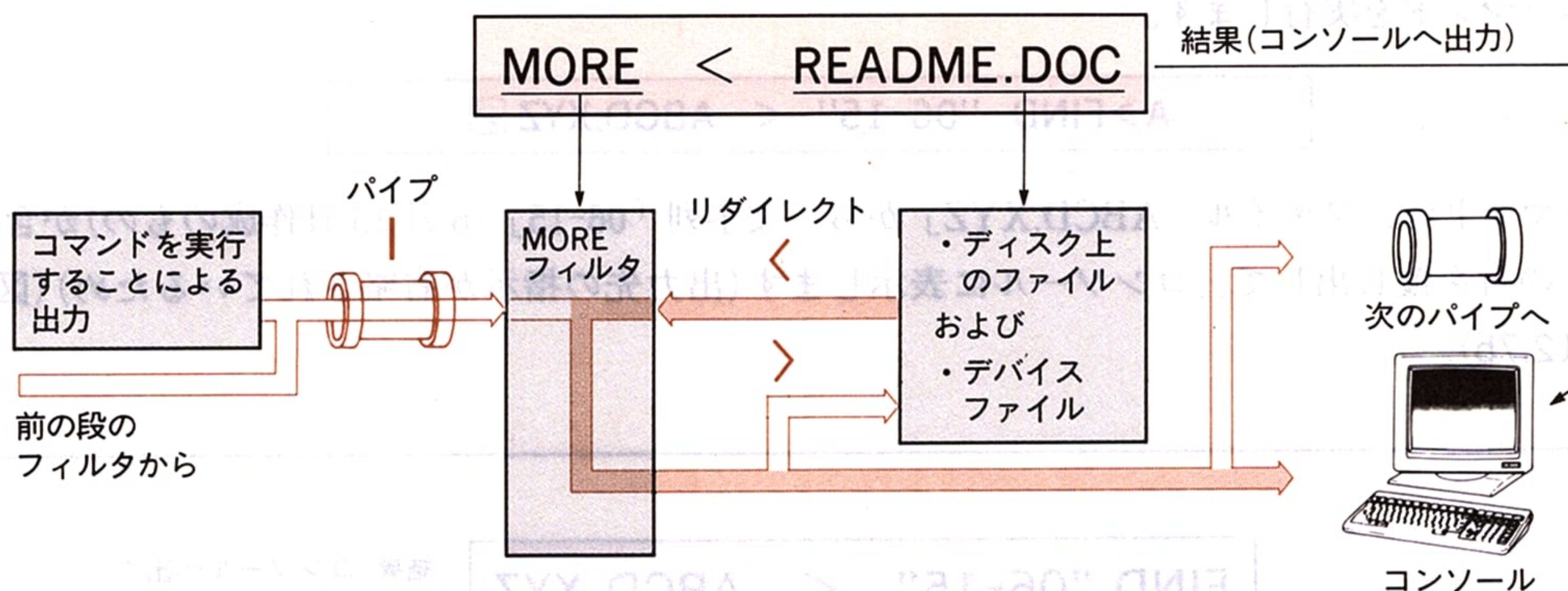



図 12.8a　ファイル README.DOC を MORE フィルタに入力し、コンソールに出力する概念図

```
A>MORE < README.DOC ↵

 1.   バックアップ作成時の注意

   PS98−013(014)−HU/H4Wのシステムディスクは、9セクタ
 フォーマット(720KB)でフォーマットされています。
 PS98−013(014)−HU/H4Wのバックアップを作成する時には、
 バックアップ媒体のフォーマット時に/9スイッチを指定してください。

 (例)

      A>FORMAT  B:  /9

   また、MENUコマンドのメニュー項目「フロッピーディスク(640KBﾀｲﾌﾟ)
 のバックアップ」を選択することにより、720KBフロッピーディスクの初期化
 からコピーまで続けて行うことができます。

   メニュー項目「フロッピーディスクのバックアップ(ドライブ指定)」は、PC
 −9801VXなどのノーマルモード機種では、メニュー画面の2ページ目にあり
 ます。
   メニュー画面のページを切り替えるには、ROLLUPキー(次のページを表示),
-- More -- x
```


図 12.8b　ファイル README.DOC を MORE フィルタに入力し、コンソールに出力する実行例

12.3 フィルタ、パイプ、リダイレクトの応用例

それでは、フィルタ、パイプ、リダイレクトを使った、いくつかの応用例を示していきましょう。

まず、次の2つの実行例は、図12.5で作成したファイル「ABCD.XYZ」を逆順にソートするものです(**図12.9a**、**図12.9b**)。それぞれの処理の方法は違っても、結果はどちらも同じになります。SORTフィルタへの入力手段であるリダイレクトとパイプの使い方の違いに注目してください。なお、実行例はすべてカレントドライブのカレントディレクトリ上で行っています。


```
A>SORT/R < ABCD.XYZ
SYS      EXE    25480  88-07-13    0:00
RSDRV    SYS     7352  88-07-13    0:00
PRINT    SYS     5855  88-07-13    0:00
MOUSE    SYS     3985  88-07-13    0:00
CONFIG   SYS       87  89-07-29   17:25
ATOK6B   SYS    21372  87-06-15   12:00
ATOK6A   SYS    54271  87-06-15   12:00
A>
```

図12.9a　ファイルABCD.XYZをリダイレクトによってSORTフィルタに入力する


```
A>TYPE ABCD.XYZ | SORT/R
SYS      EXE    25480  88-07-13    0:00
RSDRV    SYS     7352  88-07-13    0:00
PRINT    SYS     5855  88-07-13    0:00
MOUSE    SYS     3985  88-07-13    0:00
CONFIG   SYS       87  89-07-29   17:25
ATOK6B   SYS    21372  87-06-15   12:00
ATOK6A   SYS    54271  87-06-15   12:00
A>
```

図12.9b　ファイルABCD.XYZをパイプによってSORTフィルタに入力する

次の2つの実行例は、文字ファイル「README.DOC」から、「デバイス」という文字列を捜し出し、それが含まれるすべての行を表示させるものです。処理の方法は違っても、どちらも同じ結果となります。ここでもFINDフィルタへの入力手段であるリダイレクトとパイプの使い方の違いに注目してください(図12.10a、図12.10b)。

日本語入力機能で入力　ファイルであることに注目

```
A>FIND "デバイス" < README.DOC
  ＡＤＤＤＲＶ／ＤＥＬＤＲＶコマンドによって、キャラクタ系デバイスドライバ
 Ｕ）のシステムディスク＃１には、システムとデバイスドライバなどが納められて
 なデバイスドライバ，コマンドなどを集めてご使用ください。

A>
```

文字ファイル「README.DOC」からリダイレクトを使って文字列「デバイス」を捜し出す

図12.10bと同じ結果が得られる

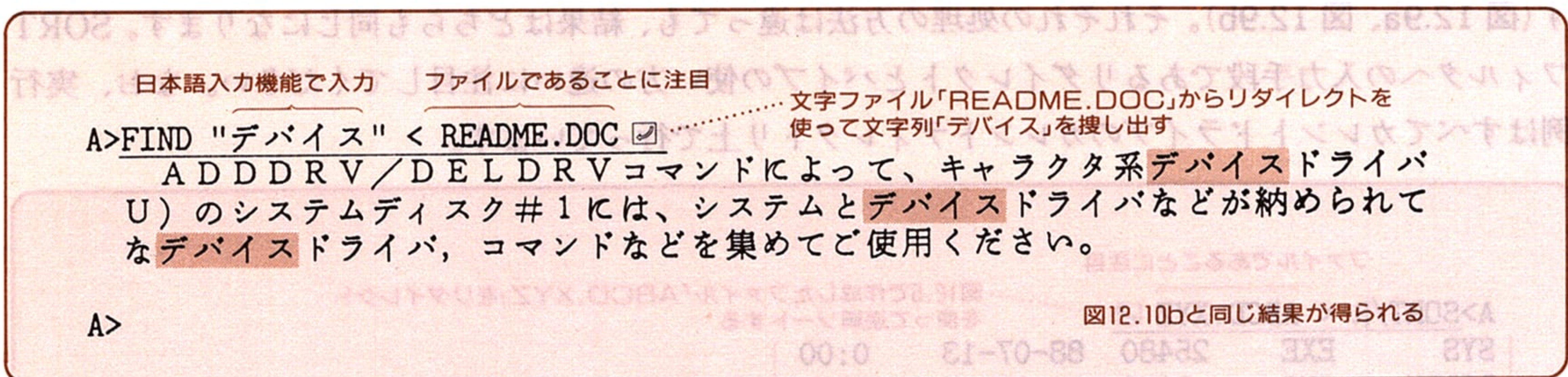



図12.10a　ファイルREADME.DOCをリダイレクトによってFINDフィルタに入力する

コマンドからの出力であることに注目

```
A>TYPE README.DOC | FIND "デバイス"
  ＡＤＤＤＲＶ／ＤＥＬＤＲＶコマンドによって、キャラクタ系デバイスドライバ
 Ｕ）のシステムディスク＃１には、システムとデバイスドライバなどが納められて
 なデバイスドライバ，コマンドなどを集めてご使用ください。

A>
```

パイプを使って、図12.10aと同じ文字列を捜し出す

図12.10aと同じ結果が得られる

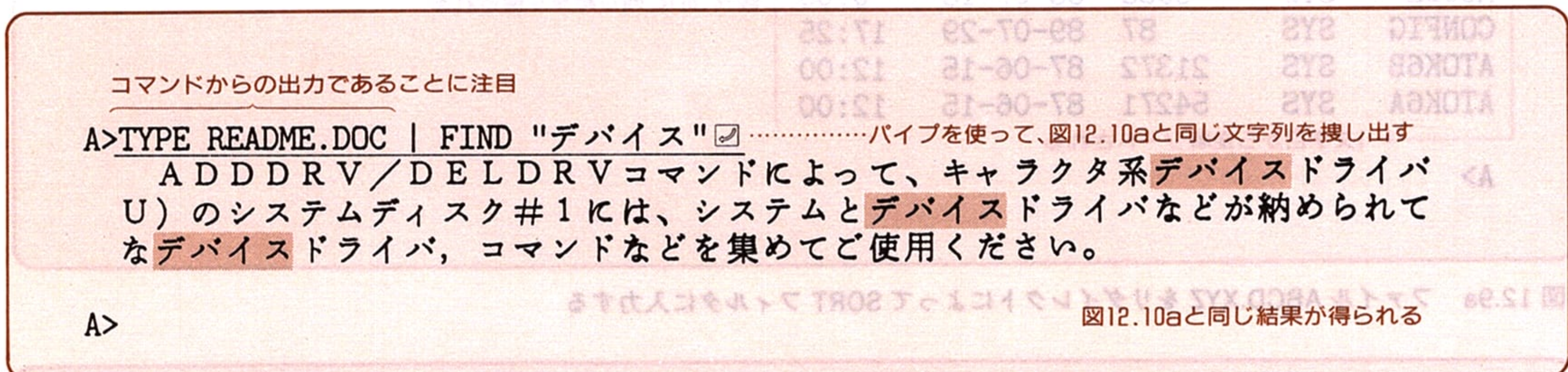



図12.10b　ファイルREADME.DOCをパイプによってFINDフィルタに入力する

次の2つの実行例は、図12.5.bで作成したファイル「ABCD.XYZ」から、文字列「06-15」つまり6月15日に作成されたファイルを捜し出して、それを逆順にソートするものです。処理の方法は違っても、どちらも同じ結果となります。コマンドライン中央のファイル名と、その左右のリダイレクトとパイプの使い方に注目してください(図12.11a、図12.11b)。

これを行った結果を ——→ これに入力する

```
A>SORT/R < ABCD.XYZ | FIND "06-15"
ATOK6B   SYS    21372  87-06-15   12:00
ATOK6A   SYS    54271  87-06-15   12:00

A>
```

図12.5で作成したファイル「ABCD.XYZ」を逆順にソートして、かつ文字列「06-15」を含む行のみをコンソールに表示する

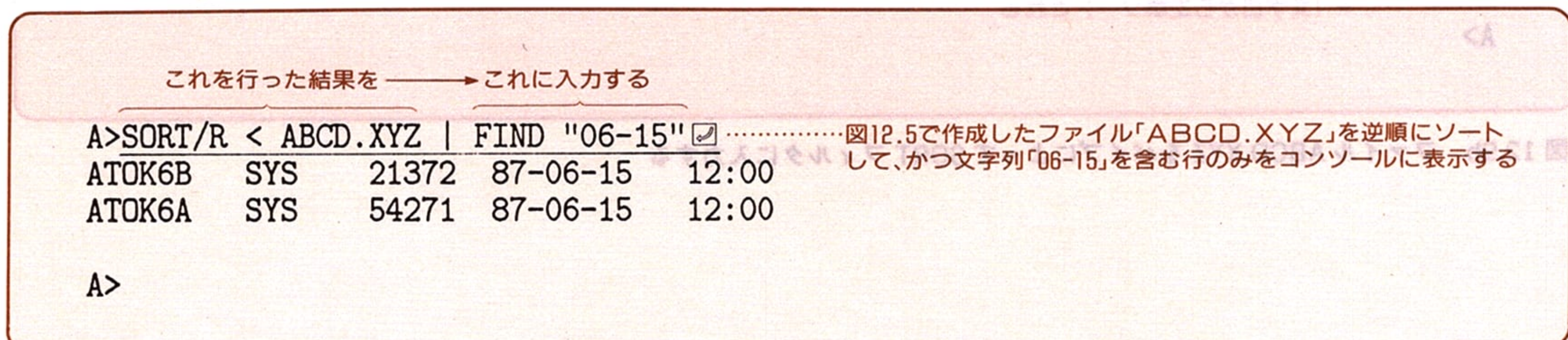



図12.11a　ファイルABCD.XYZをSORTおよびFINDフィルタに通す

```
これを行った結果を ——→ これに入力する
A>FIND "06-15" < ABCD.XYZ | SORT/R ⏎ ……図12.11bのSORTとFINDの実行の順序を逆にしたもの
ATOK6B    SYS    21372  87-06-15   12:00 } どちらを先に行っても同じ結果が得られる
ATOK6A    SYS    54271  87-06-15   12:00 }

A>
```

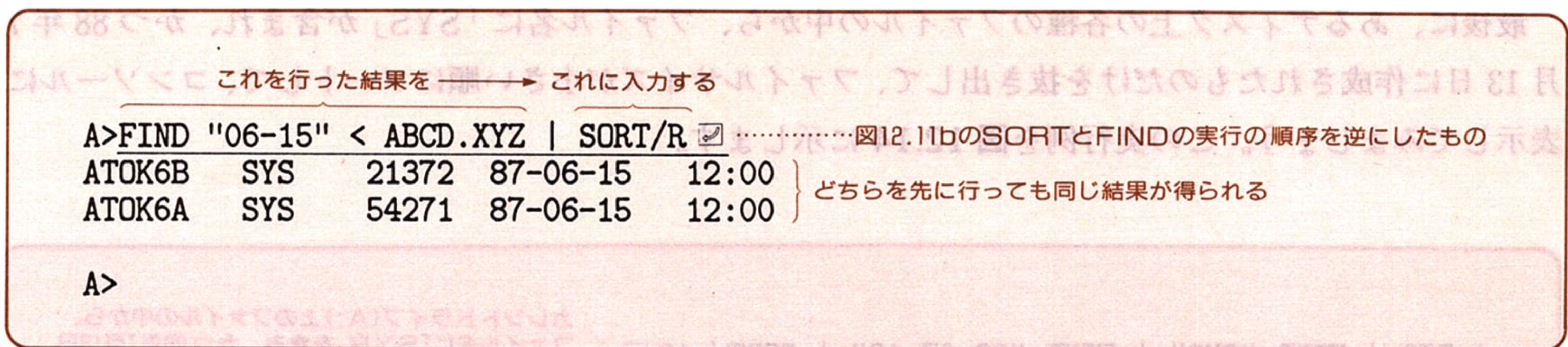



図 12.11b　ファイル ABCD.XYZ を FIND および SORT フィルタに通す

次の実行例は、DIR コマンドによる出力から、文字列「06-15」を捜し出して、コンソールに表示します。さきの 2 つの実行例とは、フィルタへ入力するデータの発生源(この例はコマンド出力、さきの例はファイル)が異なります(**図 12.12**)。

```
A>DIR | FIND "06-15" ⏎ ……カレントドライブ(A:)上のファイルから文字列「06-15」を含む行をコンソールに表示する
ATOK6A    SYS    54271  87-06-15   12:00 } 6月15日に作成されたファイルはこれだけあった
ATOK6B    SYS    21372  87-06-15   12:00 }

A>
```

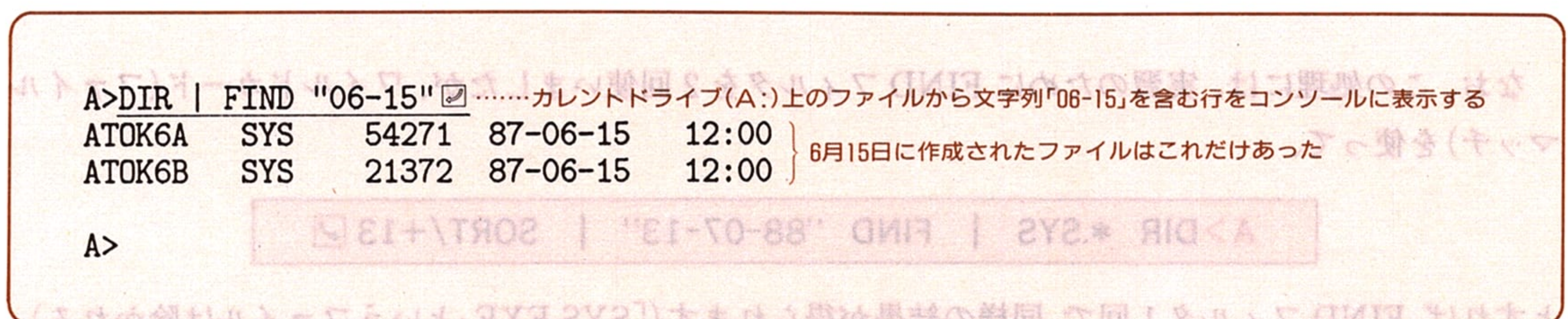



図 12.12　DIR コマンドの出力を FIND フィルタに入力する

図 12.13 の実行例は、上の例をさらに各ファイルの大きさの逆順にソートしたものです。つまり、あるディスク上の各種のファイルの中から 6 月 15 日に作成されたものを抜き出し、それをファイル容量の小さい順に並べて、コンソールに出力します。

```
A>DIR | FIND "06-15" | SORT/+13 ⏎ ……図12.12の実行例をさらに各ファイルサイズの小さい順にソートする
ATOK6B    SYS    21372  87-06-15   12:00 } 図12.12の各ファイルがサイズの小さい
ATOK6A    SYS    54271  87-06-15   12:00 } 順にソートされている
            └──→ 13文字目のここからソートの対象となる
A>
```

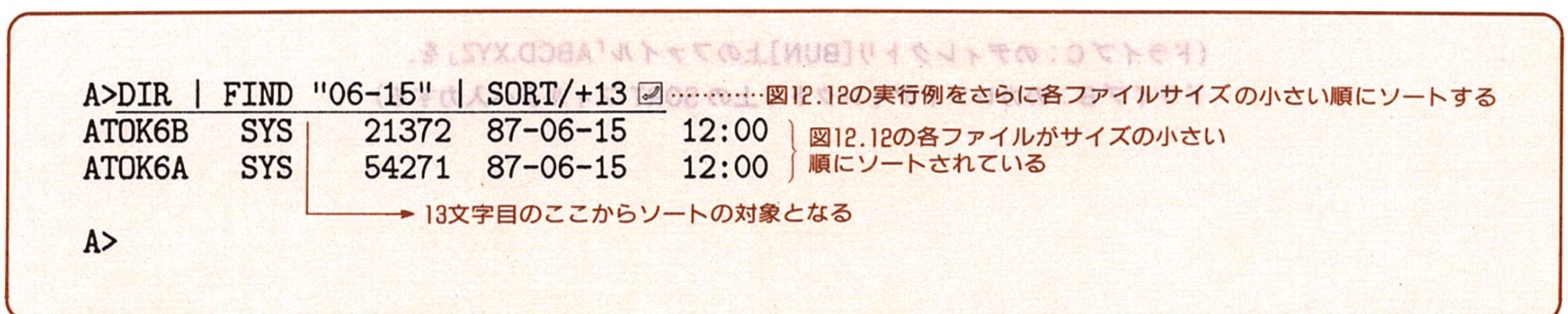



図 12.13　DIR コマンドの出力を FIND と SORT フィルタに入力する

最後に、あるディスク上の各種のファイルの中から、ファイル名に「SYS」が含まれ、かつ88年7月13日に作成されたものだけを抜き出して、ファイルサイズが小さい順にソートして、コンソールに表示してみましょう。この実行例を図12.14に示します。

```
A>DIR | FIND "SYS" | FIND "88-07-13" | SORT/+13
MOUSE    SYS     3985  88-07-13    0:00
PRINT    SYS     5855  88-07-13    0:00
RSDRV    SYS     7352  88-07-13    0:00
SYS      EXE    25480  88-07-13    0:00
A>
```

カレントドライブ(A:)上のファイルの中から、ファイル名に「SYS」を含み、かつ88年7月13日に作成されたものを捜し出し、ファイルサイズの小さい順にソートする

条件に合うものはこれだけある

13文字目からがソートの対象となる

図12.14　DIRコマンドの出力をFINDフィルタに2回、SORTフィルタに1回入力する

なお、この処理には、実習のためにFINDフィルタを2回使いましたが、ワイルドカード(ファイルマッチ)を使って、

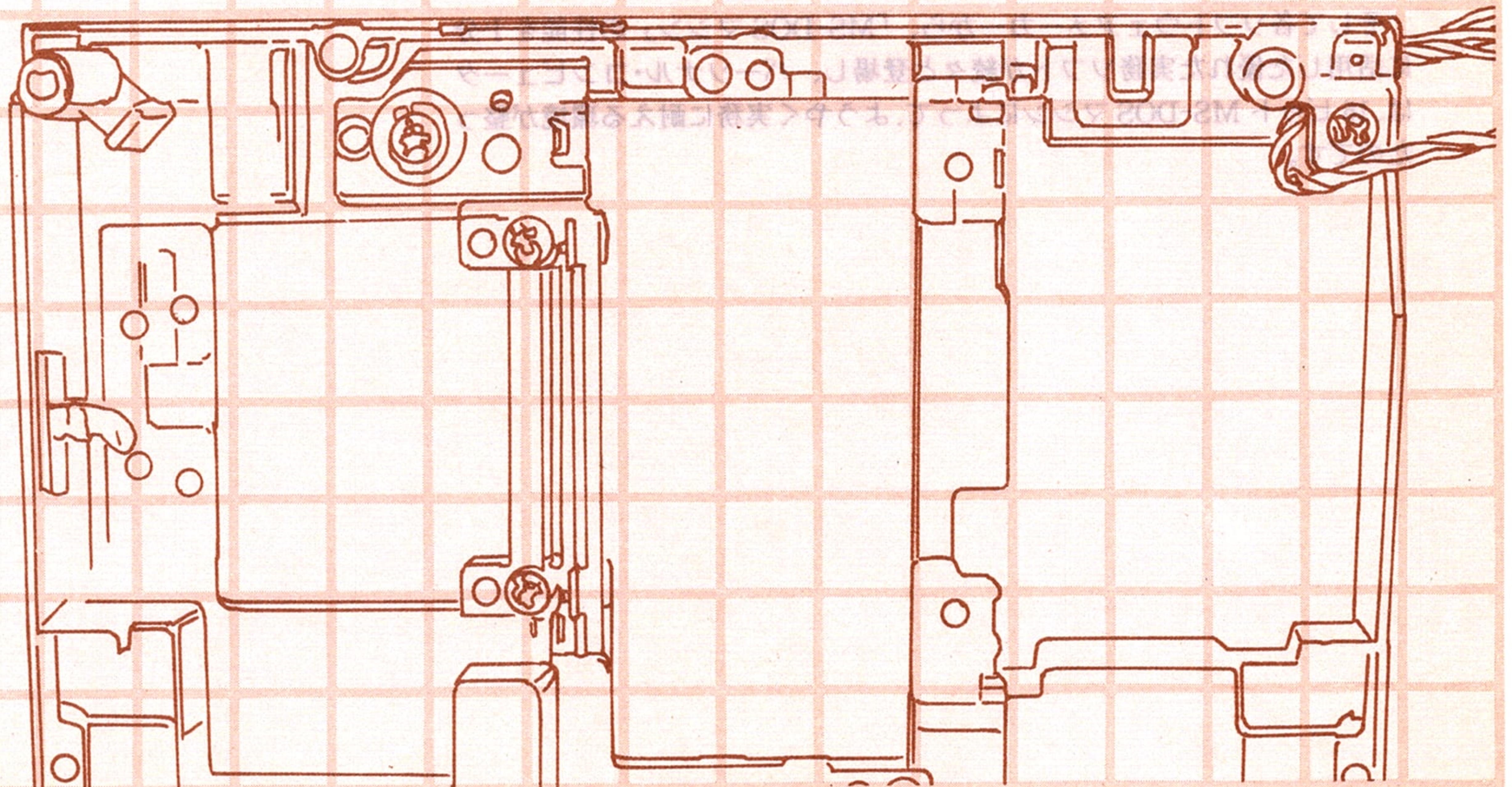
```
A>DIR *.SYS | FIND "88-07-13" | SORT/+13
```

とすれば、FINDフィルタ1回で、同様の結果が得られます(「SYS.EXE」というファイルは除かれる)。

本章での実行例は、すべてカレントドライブのカレントディレクトリ上で行いましたが、パスの指定が必要な場合は、一般の外部コマンドと同様に取り扱ってください。たとえば図12.9aの例では、次のようになります。

```
A>B:SORT/R < C:¥WP¥BUN¥ABCD.XYZ
```

(ドライブC:のディレクトリ[BUN]上のファイル「ABCD.XYZ」を、ドライブB:のカレントディレクトリ上のSORTフィルタへ入力する)

13章
MS-DOS上の各種ソフトウェア

パーソナル・コンピュータの利用分野は非常に広く、実際にどこで何に使われているか、その現状調査をすることも困難なほど、社会の隅々にまで浸透しています。その利用先の代表的なものを挙げると、

・一般ビジネス
・各種の研究開発
・各種のソフトウェア開発
・コンピュータ・グラフィックス
・CAD(コンピュータエイデッド・デザイン)
・各種の通信とそのデータ処理、および通信端末
・大型コンピュータの端末
・各種機器の制御
・各種観測装置のデータ収集
・家庭生活全般

などがあり、どれもコンピュータ利用技術の魅力ある分野といえます。本章ではこの中から、一般的に最もニーズの高い「ビジネス」と、MS-DOSに最も深く関係する「ソフトウェア開発」の分野を中心に、MS-DOSを取り巻くソフトウェア環境の一部を紹介しましょう。

パーソナル・コンピュータは、もはや私たちの社会になくてはならない存在になりました。パーソナル・コンピュータがこのように発展したのは、CPUが16ビットになり、コンピュータのハードウェア的な処理能力が大幅に向上したところに、MS-DOSという優れたバランスのOSが登場し、各社のパーソナル・コンピュータに広く採用されたことによります。

そして各ソフトウェアメーカーから、「MS-DOSマシン」の性能を十分に活用した優れた実務ソフトが続々と登場し、パーソナル・コンピュータは、16ビットMS-DOSマシンによって、ようやく実務に耐える環境が整ったのです。

13.1 ビジネスソフト

ビジネスソフトの三種の神器は、ワードプロセッサ、表計算ソフト、それにリレーショナル型データベースソフトです。この3つは正に神器であり、これらの活用次第で、一般的なビジネスのほとんどの分野をカバーすることができます。パーソナル・コンピュータによるオフィスのOA化を成功させる鍵は、この3つのビジネスソフトを、どのように使いこなすかにかかわっているといってもよいでしょう。

雑誌の広告ページを見ると、実に多くの種類のビジネスソフトが発売されていますが、実務の本格的な使用に耐えるものは、それほど多くはないものです。「やればなんとかできる」といった程度のものと、「日常の業務に使える」ものとでは、ソフトウェアの質が根本的に異なります。ここのところをよく見極めなければなりません。またビジネスソフトの場合は、各種ビジネスソフト間でのデータの互換性(交換)の問題も重要であり、それらの点にも注意が必要です。

パーソナル・コンピュータのソフトウェアについて、多くの人が誤解していると思われる点は、パーソナル・コンピュータ用の優れたビジネスソフトを開発するには、大型コンピュータ用のビジネスソフトを開発する以上に、効率のよい高度なソフトウェア技術が要求されるということです。パーソナル・コンピュータのハードウェアの能力が非力である分を、ソフトウェアによる緻密なテクニックで補わなければならないのです。

最近では、国内のソフトハウス(ソフトメーカー)も、高度な技術力を蓄えてきました。表計算ソフトや、リレーショナル型データベースソフトなどのシェアは、外国勢に圧倒されてはいるものの、国産ソフトも、ワープロソフトやカード型の簡易ソフト、それにグラフィックソフトやパソコン通信ソフトなどを中心に活況を呈しています。

13.1.1 ワードプロセッサ

ワードプロセッサ(ワープロ)はビジネスソフトの原点であり、パーソナル・コンピュータを「使うため」の原点でもあります。ワープロは、いわばビジネスソフトの「母」といってもよく、そこにはパーソナル・コンピュータを使うための、基本的な要素のほとんどすべてが含まれています。

今やワープロが自在に使えることは、ビジネスマンの常識の1つになってしまいました。その直接の目的は文書を作成することですが、それよりも「ワープロが使える」ということには、今後、パーソナル・コンピュータをいろいろなことに使っていくための大きな意味が含まれています。コンピュータに関して、これからは何をするにも、『ワープロが使える程度のパーソナル・コンピュータの基礎知識』は、すでに持ち合わせていることが前提条件であるといっても過言ではありません。

たとえば、

- ●キーボードの操作
- ●日本語入力
- ●コンピュータによるデータ処理(かな－漢字変換／各種編集処理)の概念
- ●ディスクの操作
- ●ファイルというものの概念
- ●コンピュータシステム全体の把握

など、パーソナル・コンピュータを使うための基本となる知識の多くが、ワープロを使うことによって得られるのです。

さて、ワープロソフトの発展とともに、プリンタの高性能化、高品質化もめざましく、オフィスの総合的な情報化を考えるとき、ワープロ専用機(中級・高級機)の時代はすでに終わったといってよいでしょう(ただし、低価格の簡易ワープロの需要は、「オフィスの情報化」とは別の次元のことと考えている)。最近のワープロ専用機の、ワープロ機能以外の多機能化を見れば、ワープロ専用機が、オフィスの総合情報マシンを指向していることがわかります(ワープロ専用機の中身は、パーソナル・コンピュータとほぼ同一のマシンである)。このことは、とりもなおさず、ワープロ専用機が(その正体である)パーソナル・コンピュータを指向しているわけで、このことからも、オフィスのワープロには、パーソナル・コンピュータを使うことをお勧めします。その方が人的資源を含めて、将来の発展にためにも有利でしょう。

ワープロソフトは、パーソナル・コンピュータにおける最も需要の高いソフトであることから、各社とも力を入れた、それぞれ特色のあるものが発売されています。文字と図形を統合して取り扱えるものや、デスクトップ・パブリッシング(DTP)を指向しているものなど、さまざまな種類のものがありますが、一般には、それぞれの機能がほどほどに実現できる汎用的な製品の需要が最も多く、「一太郎」「新松」といったようなものが主な製品として挙げられるでしょう。

最近の汎用的ワープロソフトの傾向として、余りにも多くの機能を詰め込みすぎ肥大してしまった反省からか、それぞれのユーザーにとって必要のない機能を、自由に取り外し可能にした製品が出てきました。同時に、ファンクションキーを含めて、それぞれのキーの機能(意味付け)を、自由に設定できるようにしたり、新しい機能を追加したり、つまり、それぞれのユーザーの使用範囲や使い方に合わせた機能を持ったワープロを、ユーザーが独自に作り上げることができるわけです。

ワープロソフトの選択には、もう1つ重要な要素があります。ワープロソフトを選択する際には、それぞれのワープロソフトに使われている「日本語入力システム」(FEP：フロントエンド・プロセッサと呼ばれている)を、ほかのビジネスソフトにも共通に使うことを考えなければなりません。一般に、ワープロ以外のビジネスソフトやその他のソフトにも、自分が使っているワープロと同じ日本語入力

一太郎
販売／(株)ジャストシステム

新松
販売／(株)管理工学研究所

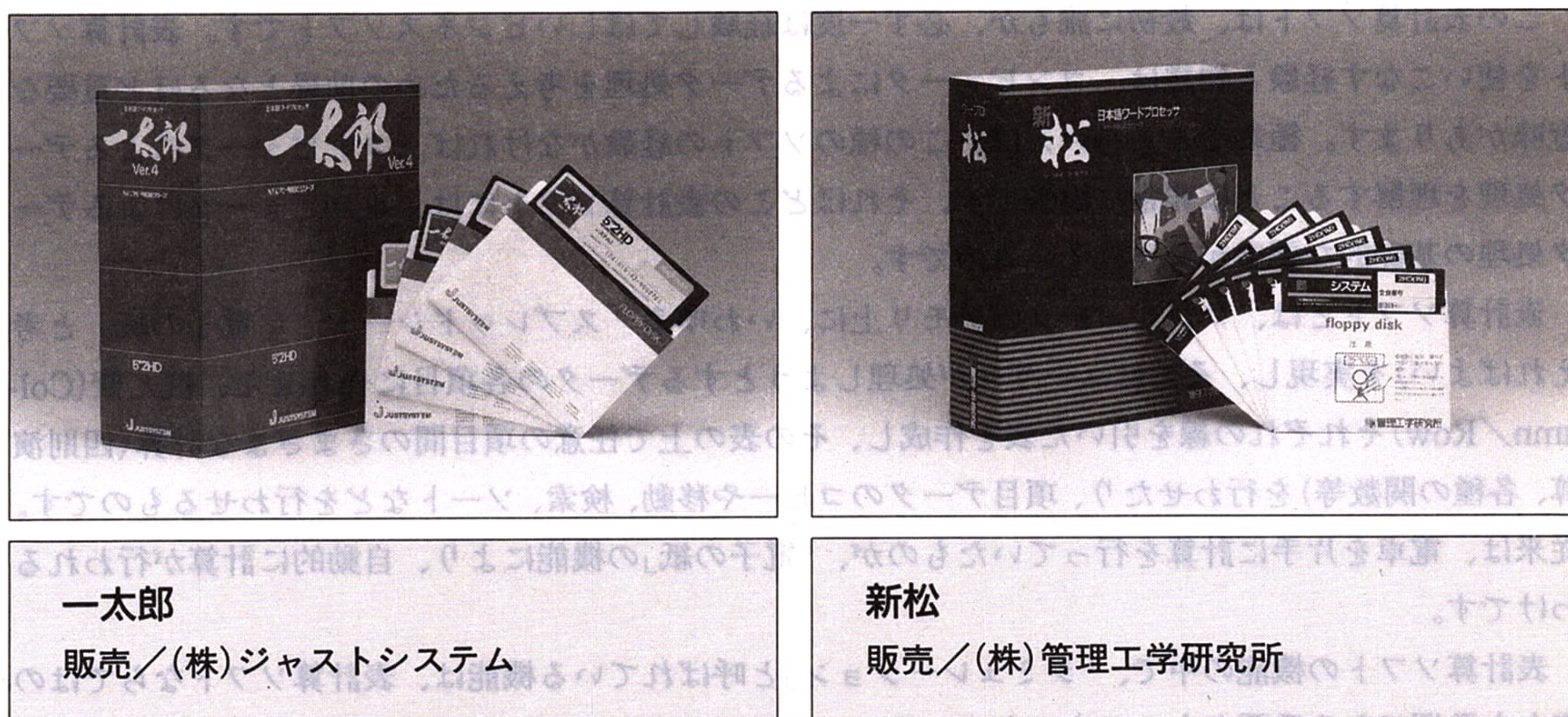



写真 13.1　MS-DOS 上の代表的なワープロソフト

システムで日本語を入力できることが理想です。そこで、ワープロソフトの選択は、「その日本語入力システムを、すべてのアプリケーションソフトにも使用する」ということを前提に検討する必要があります。

いずれにしても、ワープロの選択には、仕事の種類、目的、使う人の条件などを考慮して、それぞれに適した製品をよく調査する必要があります。なお、ワープロの「使い心地」は、フロッピーディスクで実行する場合と、ハードディスクや RAM ディスク(メモリ用の IC で実現した高速のディスク装置)、ディスクキャッシュ(増設 RAM などを利用して、ディスクの読み出しを高速化する手法)などを使って実行する場合とでは、動作スピードに格段の差が出ます。用途によってハードウェアシステムの検討も必要です。

13.1.2　表計算ソフト

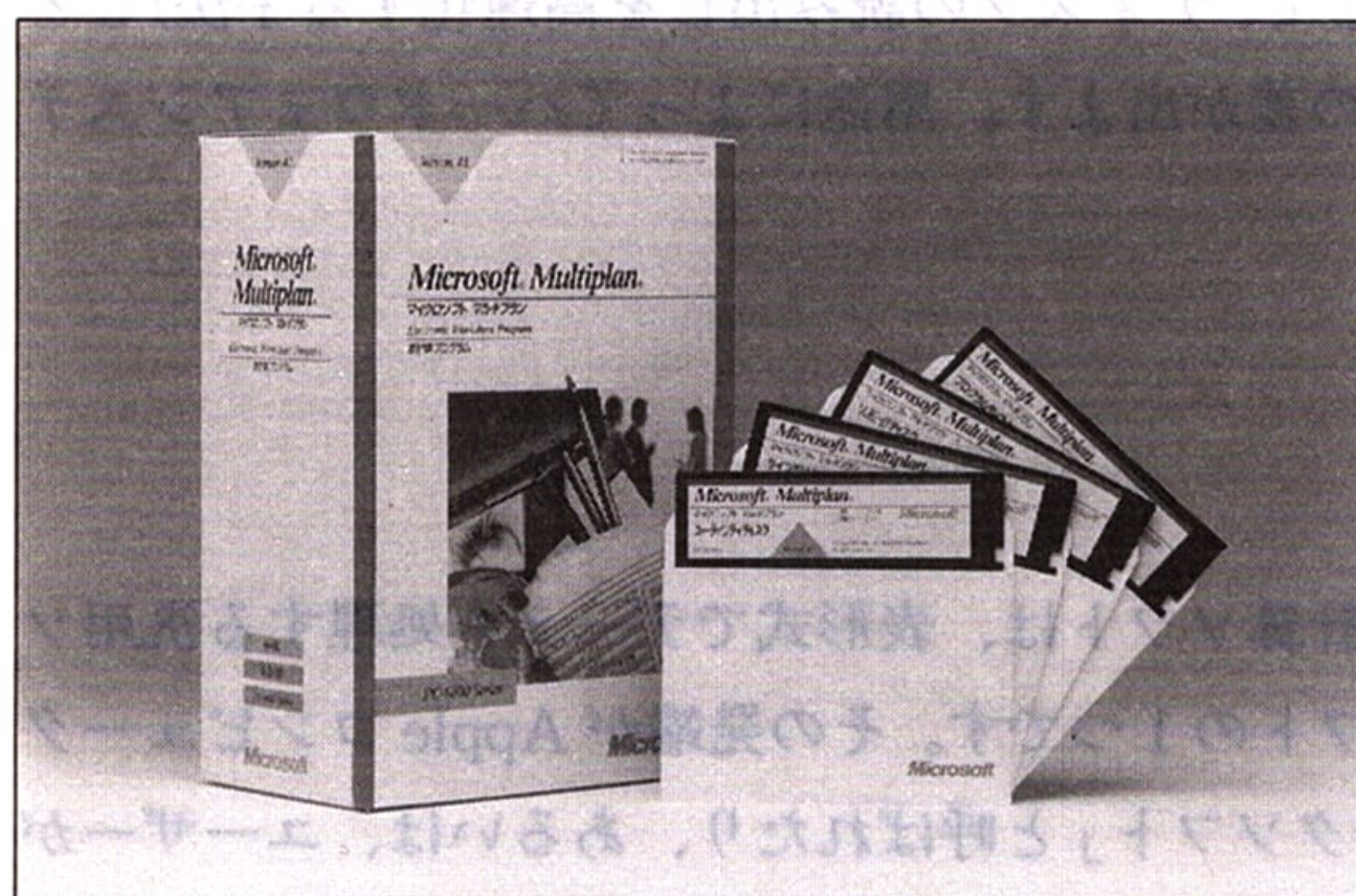
「Multiplan」や「Lotus 1-2-3」に代表される表計算ソフトは、表形式でデータを処理する汎用ソフトであり、現在最も広く使われているビジネスソフトの1つです。その発端が Apple コンピュータ用の「VisiCalc」(1978 年に登場)であるため、「カルクソフト」と呼ばれたり、あるいは、ユーザーがプログラムを組む必要がないために、「プログラムレス言語」とか「簡易言語」などとも呼ばれています。

この種のソフトは、動作が直感的でわかりやすく、誰もが容易に操作できるため、さまざまなビジネスデータの処理に幅広く利用されています。もちろん「ビジネスデータ」に限らず、あらゆる分野のデータの処理に応用できることはいうまでもありません。

この表計算ソフトは、最初に誰もが、必ず一度は経験してほしいビジネスソフトです。表計算ソフトを使いこなす経験と知識は、コンピュータによるデータ処理を考えるための前提となるほど重要な意味があります。極端なことをいえば、この種のソフトの経験がなければ、コンピュータによるデータ処理を理解することは非常に困難です。それほどこの表計算ソフトには、コンピュータによるデータ処理の基礎的な要素が含まれているのです。

表計算ソフトとは、コンピュータのメモリ上に、いわゆる「スプレッドシート」(「電子の紙」と考えればよい)を実現し、そこにユーザーが処理しようとするデータの各項目に合わせた、縦／横(Column／Row)それぞれの線を引いた表を作成し、その表の上で任意の項目間のさまざまな計算(四則演算、各種の関数等)を行わせたり、項目データのコピーや移動、検索、ソートなどを行わせるものです。従来は、電卓を片手に計算を行っていたものが、「電子の紙」の機能により、自動的に計算が行われるわけです。

表計算ソフトの機能の中で、「シミュレーション」と呼ばれている機能は、表計算ソフトならではの威力を発揮できる重要なものです。あるパラメータ(値)を変化させることにより、それに関連するさまざまなデータの変化を、その場で直ちに算出･表示すること(これをシミュレーションと呼ぶ)は、表計算ソフトでなければできません。また、表計算ソフトで作成されたデータを、グラフ表示ソフトに入力して各種のグラフを描かせたり、ワープロやほかのビジネスソフト間で、互いのデータの交換を行ったりすることも可能です。

ビジネスデータ処理の原点は、この表計算ソフトです。データ処理を考えるには、まずこのソフトを使いこなすことが先決です。そしてこのソフトを使っていくうちに、この種のソフトでは実現でき

Multiplan
販売／マイクロソフト(株)

Lotus 1-2-3
販売／ロータス(株)

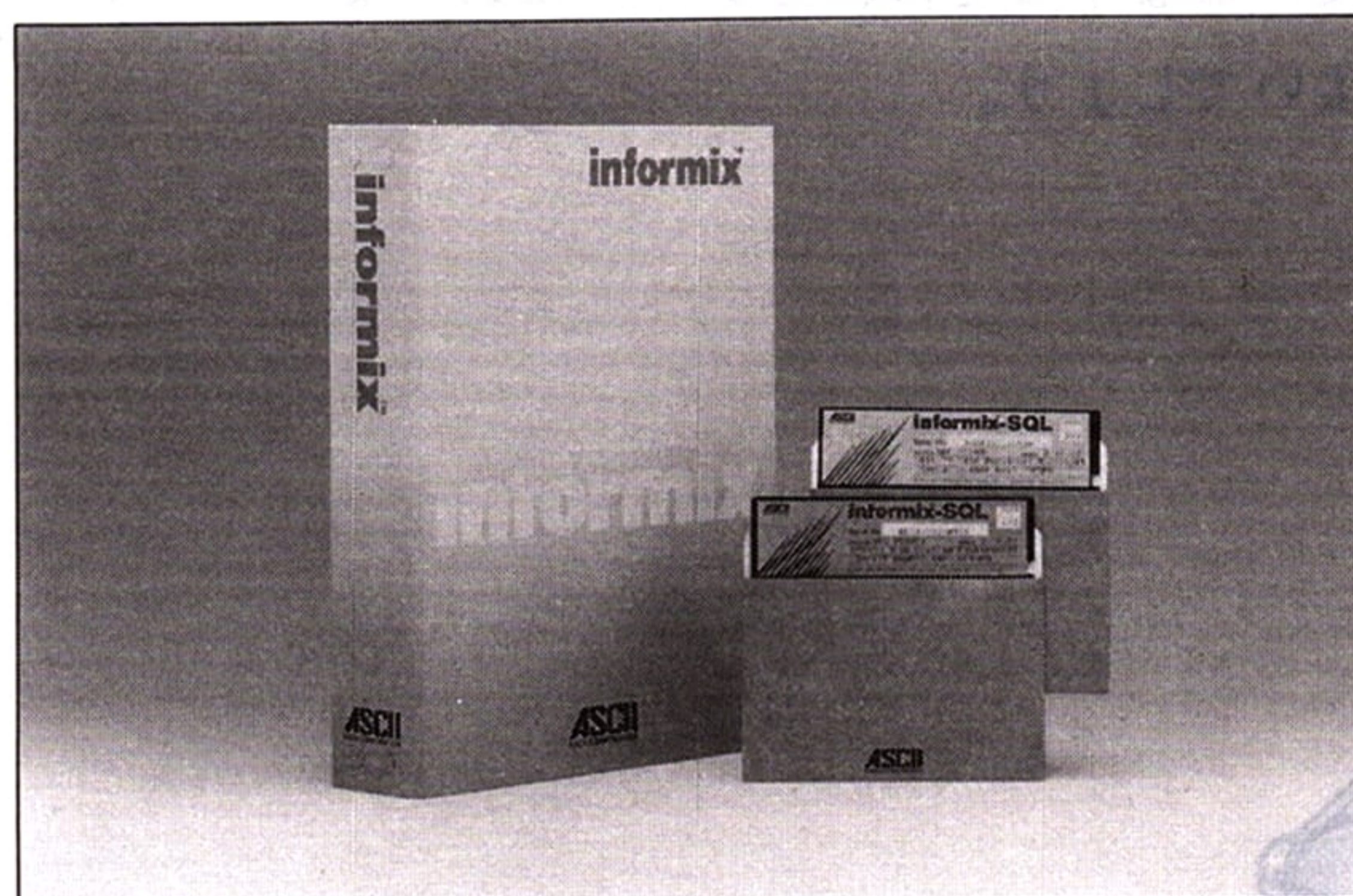
写真 13.2　MS-DOS 上の代表的な表計算ソフト

ない、もっと高度な処理の要求が出てくるでしょう。そのような要求が出てくることこそが重要なポイントであり、そこが本格的なビジネスソフトの本命、リレーショナル型データベースに取り組むための出発点になるのです。

パーソナル・コンピュータを、ビジネスデータの処理に利用しようと考えている方は、まずはじめに、ぜひこの表計算ソフトを経験しておくことをお勧めします。

13.1.3　データベース

ビジネスソフトの本命は、リレーショナル型データベースです。表計算ソフトは、基本的には1つの表の縦横の平面的な計算や並べ換えを行うだけですが、リレーショナル型データベースはそれらの機能に加え、同じ項目が含まれる複数の表の中から、必要なデータをいろいろな条件によって結びつけたり、1つの表の各項目にそれぞれの条件をつけ、そのすべての条件に合致するデータのみを取り出したりすることなどが可能です。つまり、データ処理に関する、ほとんどすべての基本的機能を持っているわけです。表計算ソフトで処理が可能な比較的単純な作業で、かつその処理だけで完結するような仕事以外は、このリレーショナル型データベースを利用することになるでしょう。結局、本格的な業務の多くは、ほとんどがこのソフトを必要とします。またこのソフトによってこそ、ビジネスのデータ処理におけるコンピュータの力がフルに発揮されることにもなるのです。

前項で、表計算ソフトを経験することが、リレーショナル型データベースを理解するための基礎になることを述べましたが、リレーショナル型データベースのもとになるデータは、やはり表計算ソフ

Informix-SQL
販売／(株)アスキー

桐
販売／(株)管理工学研究所

写真 13.3　MS-DOS 上の代表的なリレーショナル型データベースソフト

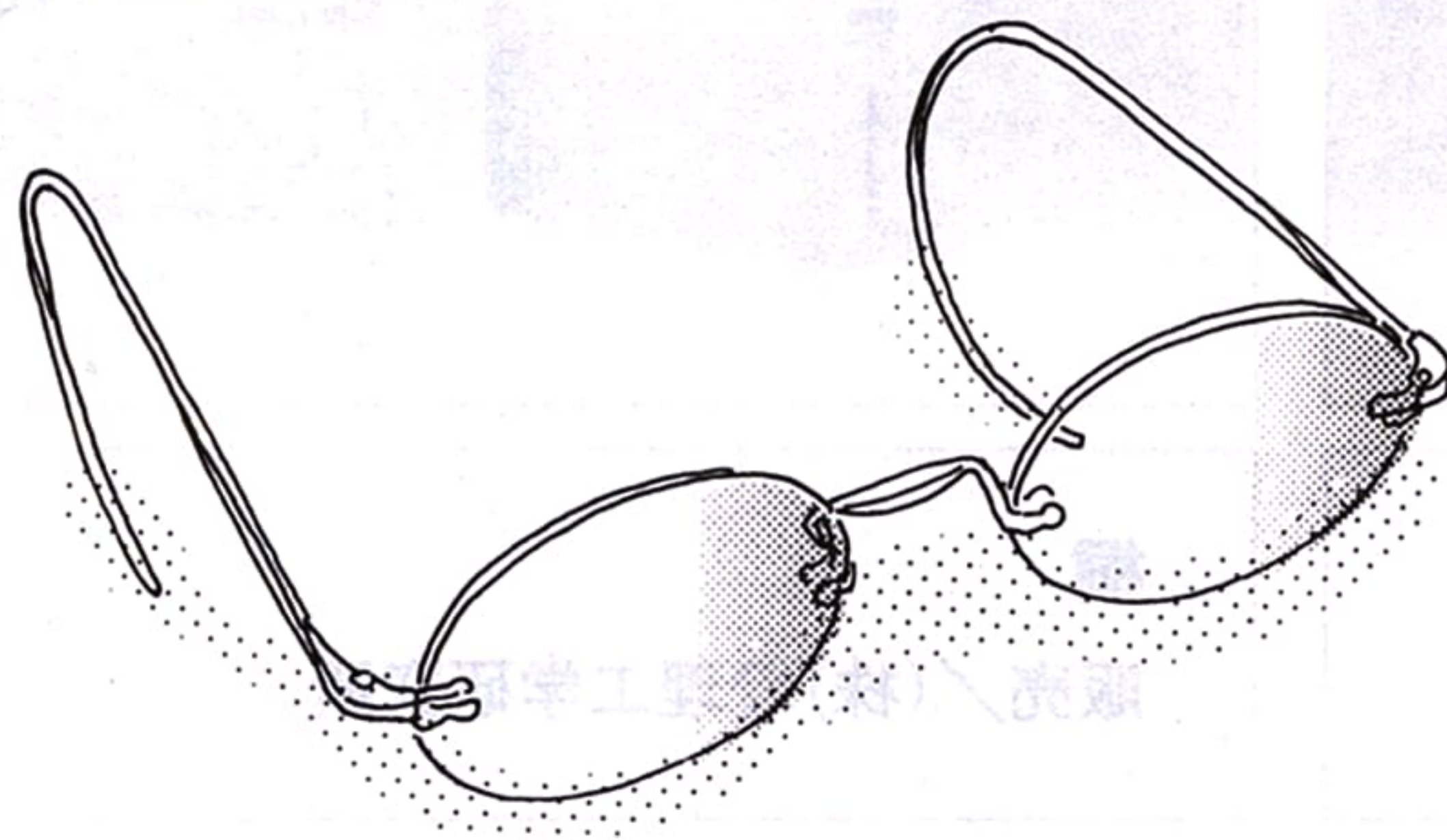
トのような、縦／横の「表」なのです。この1つの「表」のことをリレーショナル型データベースの用語では、ファイルとかリレーションと呼び、ある同じ項目のデータを縦にながめたものを「フィールド」、全項目がそろった1組のデータを横にながめたものを「レコード」などと呼んでいます。

表計算ソフトでは、表がすべてであり、基本的には1つの表の上でデータの計算や、並べ換えを行いますが、リレーショナル型データベースでは、表(ファイル)はいろいろな処理を行うための入力データの1つにすぎないわけです。リレーショナル型データベースが行う基本的な処理は次の3つです。

- ファイルの中から、あるいくつかの条件を満足するレコードを抜き出して、新しいファイルを作成する(条件抽出)
- ファイルの中から、必要ないくつかのフィールドだけを取り出して、新しいファイルを作成する(項目抽出)
- 複数のファイルから、関連するフィールドをもとに、ほかの必要なフィールドを結合して、新しいファイルを作成する(ファイル結合)

このような基本的な処理を総合的に活用し、データベース化(一元管理化)された各種のデータをさまざまに処理するソフトが、リレーショナル型データベースです。

その他の付加的な機能として、データの入力画面や入力方法、出力画面やその形式などを任意に設定したり、プリントアウトの形式を、それぞれ任意に設定することなどが可能です。つまり、本来は汎用であるリレーショナル型データベースソフトを、それぞれの仕事に合わせた専用ソフトウェアに作り上げる(カスタマイズする)ことができるのです。

現在のリレーショナル型データベースソフトは、ユーザーインターフェイスの面(操作性の面)では、まだまだ改善する必要はあるものの、このソフトを使いこなすことは、ビジネスにおけるコンピュータの本当の力を手に入れることに等しいといってもよいでしょう。

13.2 ソフト開発言語

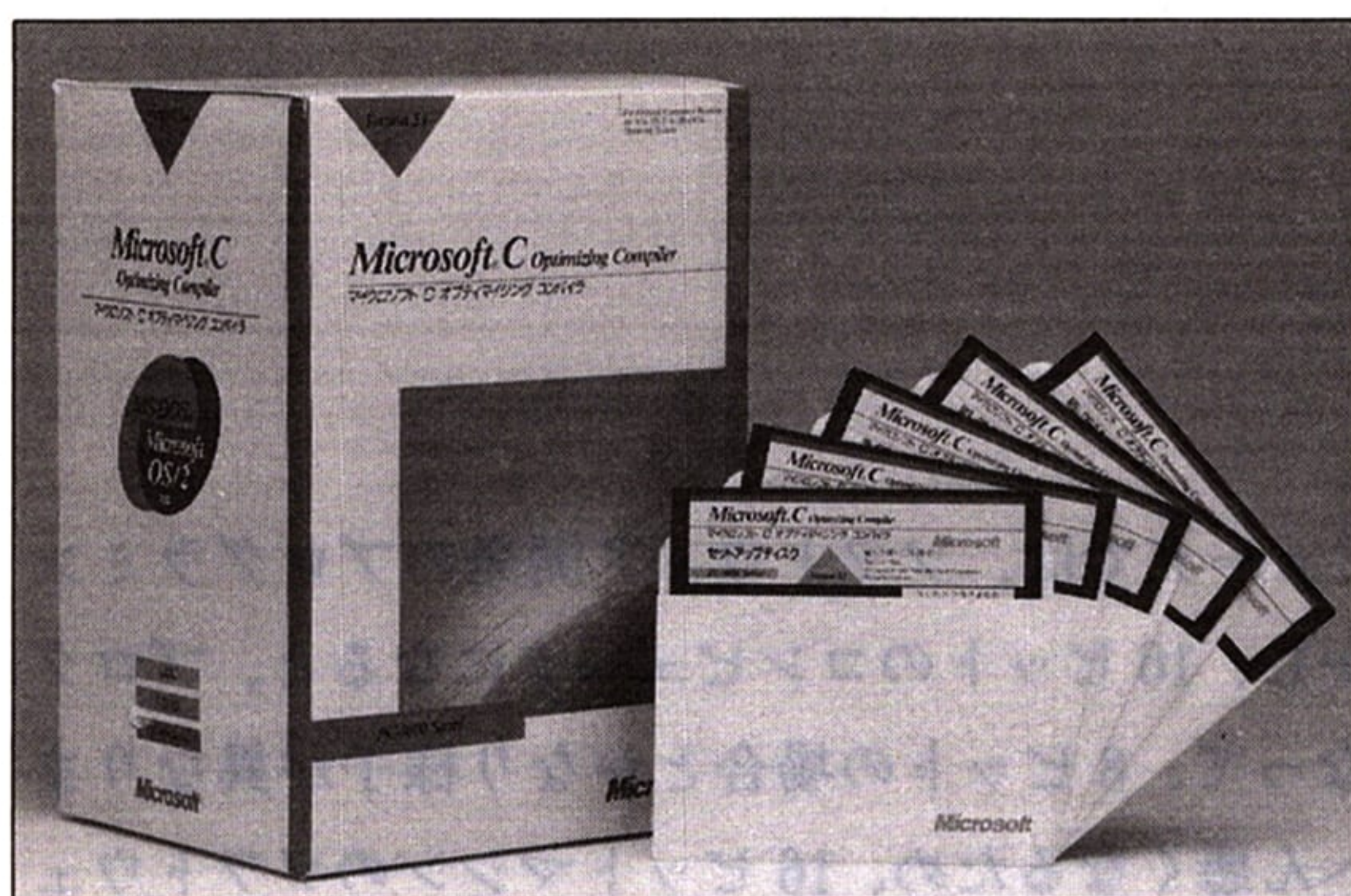
13.2.1 プログラミング言語

Z-80や8085などの8ビットのCPUを持つコンピュータでは、ソフトウェア開発のプログラミング言語としてアセンブラが多く使われています。しかし、16ビットのコンピュータになると、プログラムの規模も大きくなり、アセンブラ自体も複雑になって、8ビットの場合とかなり様子が異なります。アセンブラを使っていたのでは開発効率がたいへん悪くなるため、16ビットマシンのソフトウェア開発には、ほとんどの場合、いわゆる「高級言語」と呼ばれるプログラミング言語が使われます。もちろん部分的には、アセンブラを併用することもありますが、ほとんどを高級言語に頼って開発が行われているのが実状です。

高級言語といっても、マシンに組み込まれているBASICのことではありません。本格的なプログラミングには、「インタープリタ」形式のBASICは使われません。最終的にマシン語が生成される「コンパイラ」と呼ばれる形式のプログラミング言語が使われます。インタープリタ形式というのは、マシン語を作成するのではなく、ユーザープログラムを実行する時点で、ユーザーが書いたプログラムの1行ごとを逐次マシン語に変換しながら実行して行く形式のもので、マシンに組み込みのBASICがその代表例です。

MS-DOS上の各種プログラミング言語の種類は、ないものはないといってもよいほど豊富にそろっています。価格も安価になってきました。その中で、従来から主流の座にあるプログラミング言語は「C」です。C言語は、MS-DOSをはじめとするパーソナル・コンピュータのOSが指向する「UNIX」の標準プログラミング言語であり、その機能、使いやすさ、移植性の良さなどから、ソフトハウスや各大学、研究所を中心に急速に普及したもので、現在のMS-DOS上のアプリケーションソフトの多くは、このC言語によって書かれています。また、このC言語の登場がなかったら、現在のようなアプリケーションソフトの発展――ひいてはパーソナル・コンピュータの発展はなかったのではないかとさえ思われます。

最近のプログラミング言語ソフトの製品の発展はめざましく、「エディット→コンパイル→デバッグ→再びエディット」というように、ソフト開発全体の作業環境をトータルで考え、エディタからデバッガまで、それぞれのツールを1つの言語製品にまとめたものが増えてきました(Quick C、Turbo C、Quick BASICなど)。これらのプログラミング言語を使うことにより、開発過程の各作業間の連係が非常にスムーズに行え、開発効率を上げることができます。さらに、シェアの高い機種に対応した製品は、その機種のグラフィック機能をサポートしているものもあり、それらを使えば、グラフィック関係のソフトの開発も非常に容易になります。

なお、C言語によるソフト開発の実例は、本書の続巻(応用編)で紹介します。

Microsoft C
販売／マイクロソフト(株)

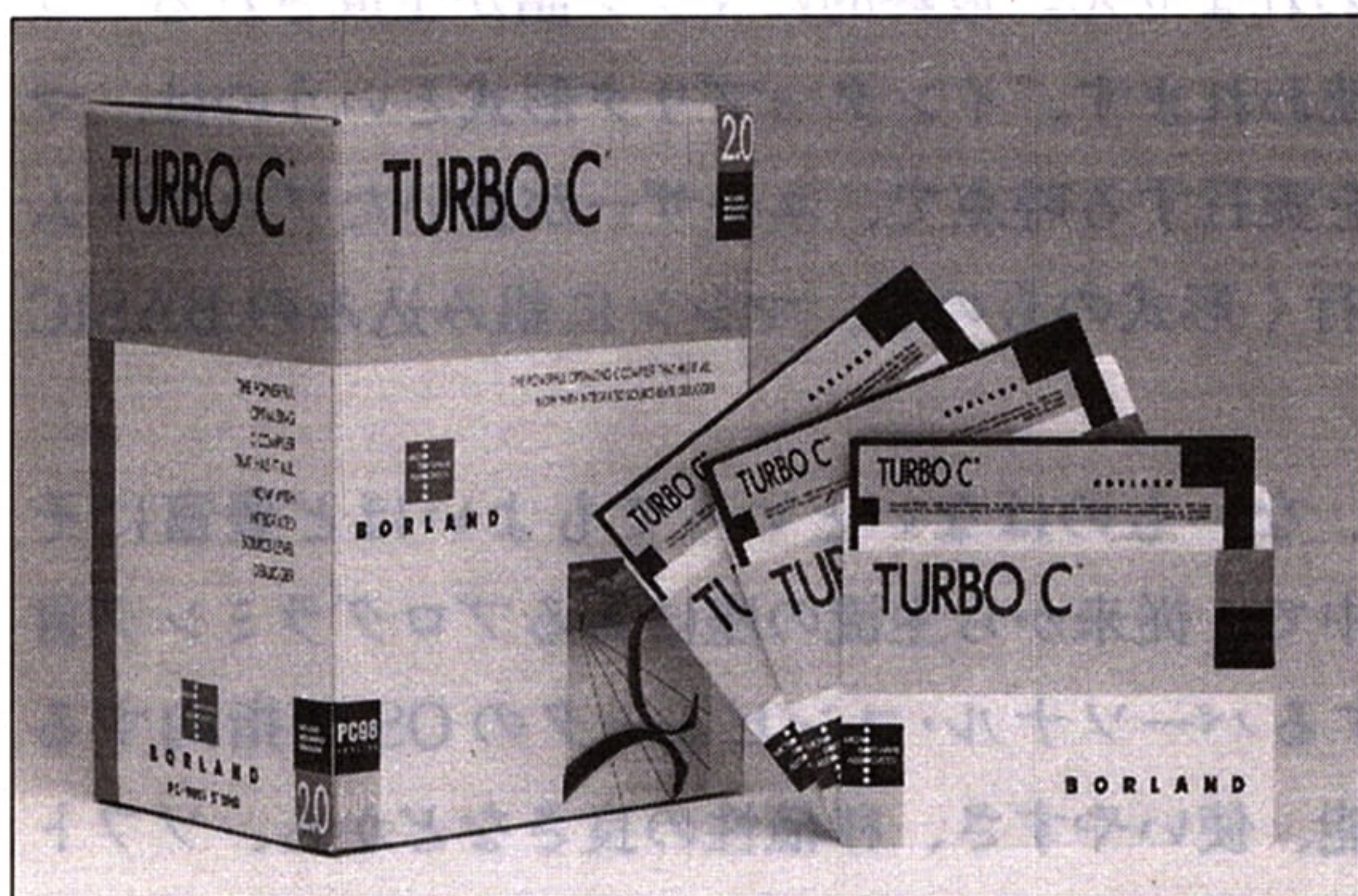
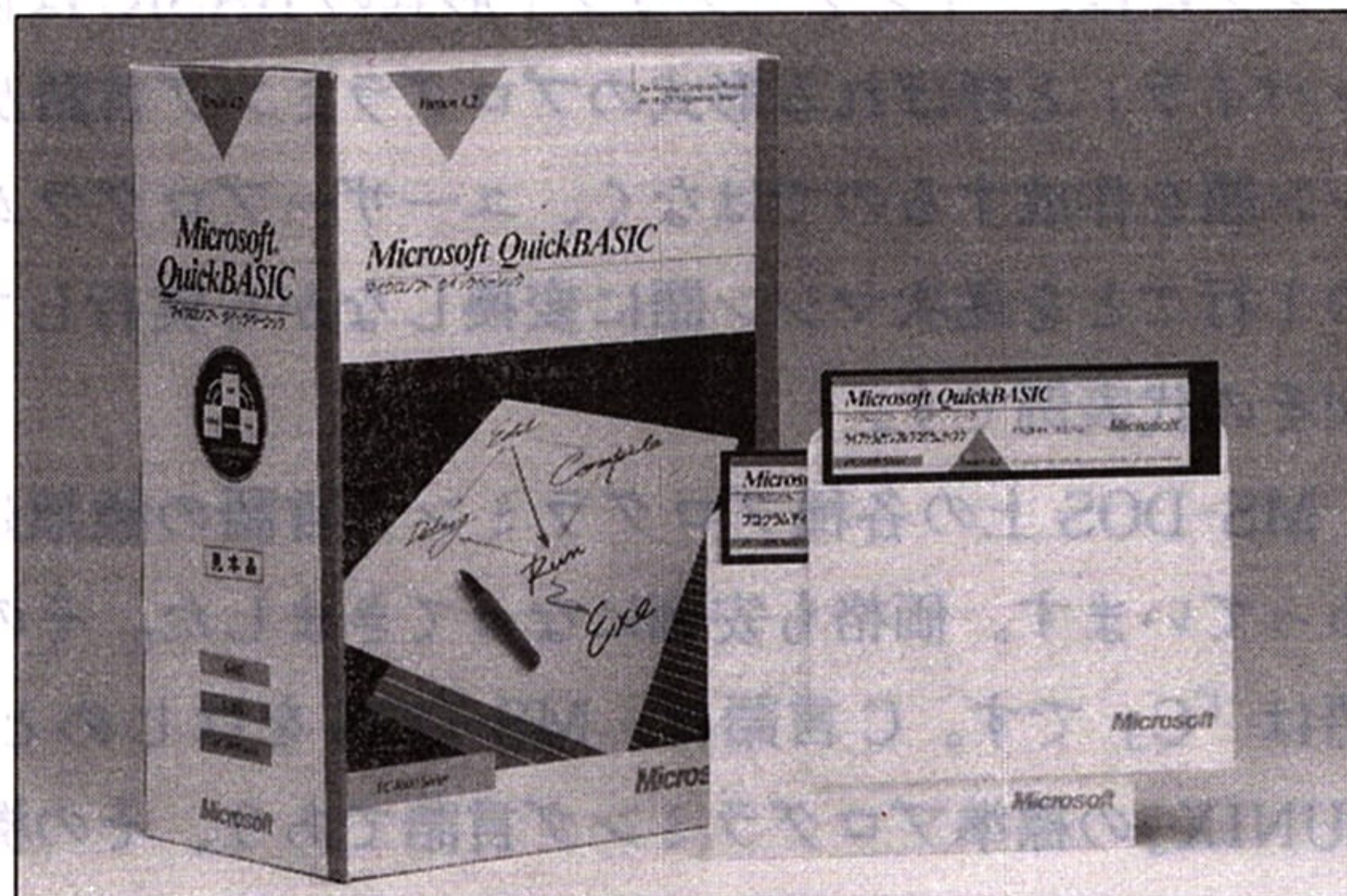
Quick C
販売／マイクロソフト(株)

Turbo C
販売／(株)マイクロソフトウェアアソシエイツ

Quick BASIC
販売／マイクロソフト(株)

写真13.4　MS-DOS上の代表的なプログラミング言語ソフト

13.2.2　エディタ

各種のプログラミング言語のソースプログラムを書いたり、さまざまな文字ファイルを作成するには、各種の編集機能が充実しているエディタが必要になります。ワープロは、同じ文字ファイルを作成できるソフトではありますが、ビジネス文書などの作成には適しているものの、編集機能の豊富さの点で、いわゆる「エディタ」には及ばず、やはり用途を異にしています。

MS-DOSのシステムディスクには、EDLINというエディタが標準装備されていますが、このEDLINは、ポインタ形式のラインエディタであり、けっして使いやすいものではありません。そこでソフト開発者の多くは、各メーカーから発売されている操作性のよいエディタ(スクリーンエディタ形式のもの)を使うことになります。

最近のエディタソフトの発展はめざましく、「MIFES」「RED」などといった製品の多くが、動作の高速性、自由なキーアサイン(それぞれのキーの機能を任意に設定すること)、マクロ機能(ユーザーの求める独自の機能をプログラムしてエディタに組み込む機能)、マルチウィンドウによる複数ファイルのエディット、1行の文字数制限の撤廃、それにエディタ自身がコンパクトであること(100Kバイト程度)などのさまざまな機能を実現しています。

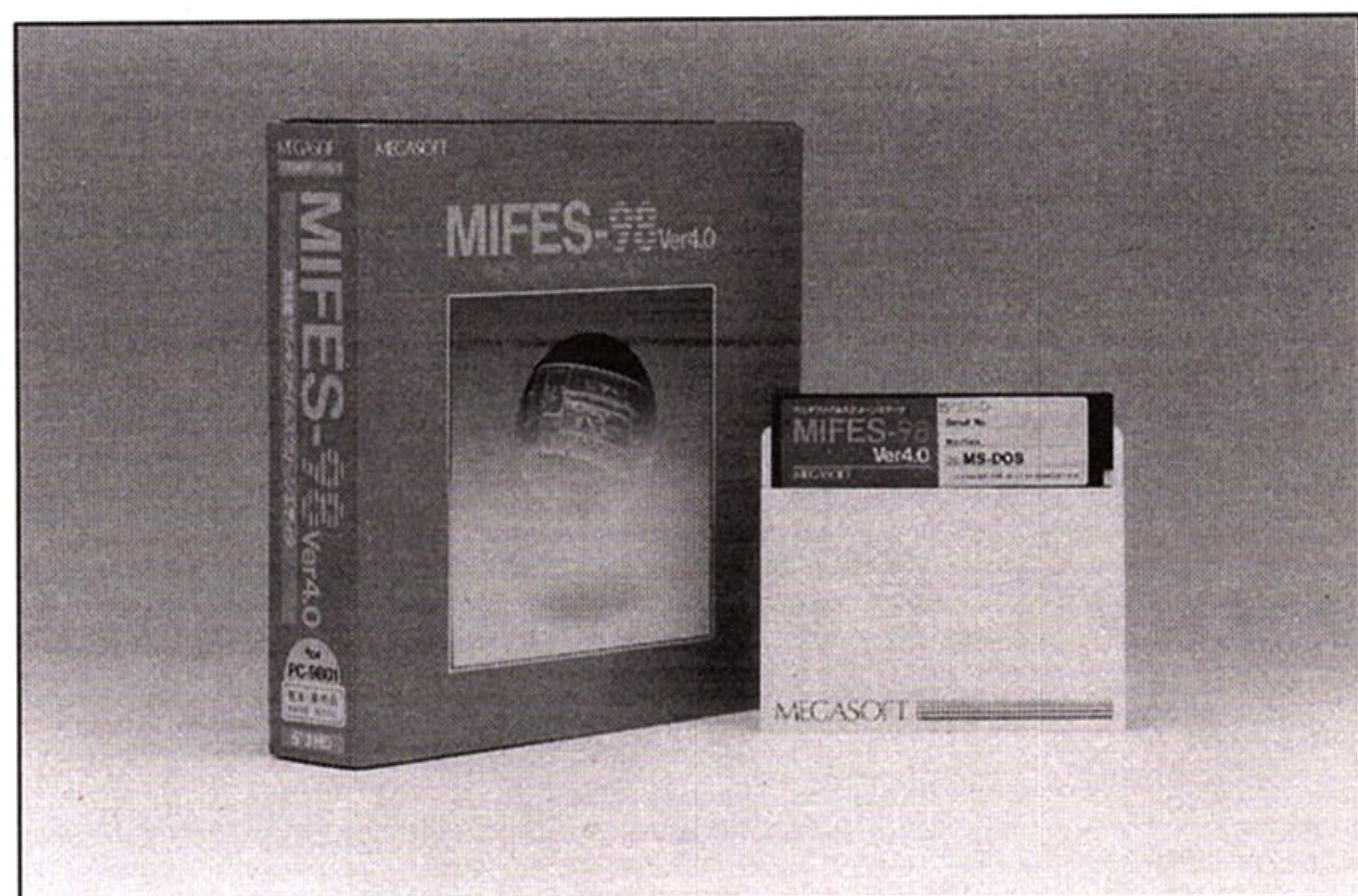
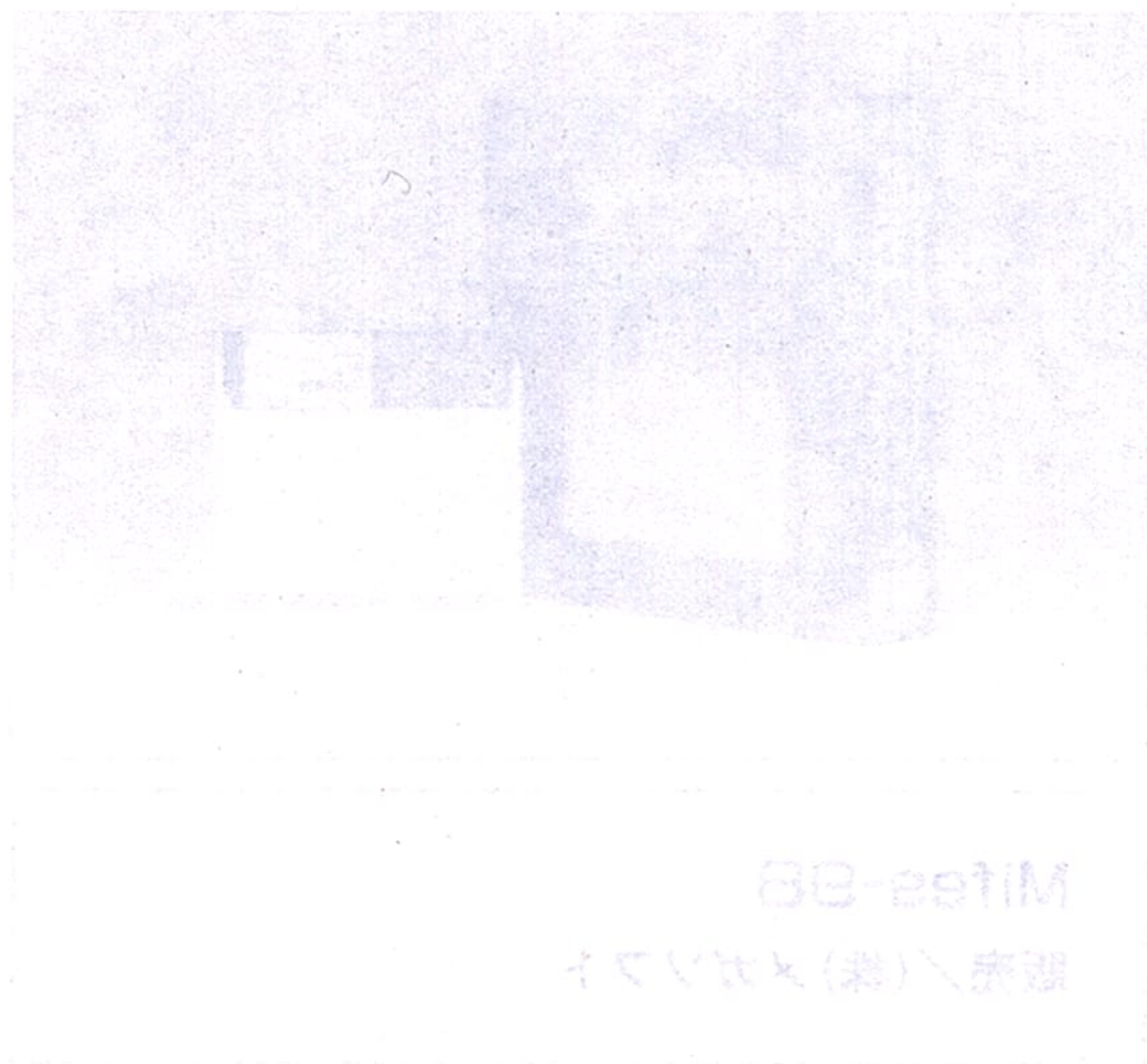
なお、エディタの代用として、使い慣れたワープロを簡易的に使うこともできます。たいていのワープロには、入力した文書を、各種プログラミング言語のソースプログラムとして使える「文字ファイル形式」(テキストファイル形式)で出力する機能がついていますので、作成した文書をディスクに保存する際に、このテキスト形式を選択すればよいわけです。この実例や、エディタ(Mifes-98、EDLIN)の使い方などは、本書の続巻『実用MS-DOS』で紹介します。

Mifes-98
販売／(株)メガソフト

写真 13.5　MS-DOS上の代表的なエディタ

本章で取り上げたソフトウェアを快く提供していただいた、

(株)アスキー
(株)管理工学研究所
(株)ジャストシステム
マイクロソフト(株)
(株)マイクロソフトウェアアソシエイツ
(株)メガソフト
ロータス(株)

の各社にお礼を申し上げます。

APPENDIX

CONFIG.SYS ファイルとデバイスドライバ

本書の所々(図 4.5b、4.6、6.1 など)に、**CONFIG.SYS** というファイル名が登場しています。このファイルは、各種の日本語入力システムをはじめ、RAM ディスクとか、ディスクキャッシュ、EMS メモリ、それに PC-9800 シリーズの MS-DOS バージョン 3.*x* では、プリンタや RS-232C インターフェイスなどを動作させるためのプログラムを、MS-DOS システムに組み込むためなどに必須の重要なファイルです。CONFIG.SYS ファイルは、**コンフィギュレーションファイル**とか**システム構築用ファイル**などと呼ばれていますが、その内容の一例を図 **A.1** に示しましょう。

```
A>TYPE CONFIG.SYS ⏎ ………TYPEコマンドでCONFIG.SYSファイルの内容を表示する
FILES = 20      ⎫
BUFFERS = 20    ⎭ MS-DOSの動作環境を設定するための指示
DEVICE = ATOK6A.SYS /E=1 /T=1 ⎫
DEVICE = ATOK6B.SYS           ⎭ ATOKをMS-DOSシステムに組み込むための指示

A>
```

図 **A.1**　**CONFIG.SYS ファイルの一例**

これは、Multiplan の常用ディスクにおける CONFIG.SYS ファイルの内容(最も簡単な例)を、TYPE コマンドで表示したものです。この CONFIG.SYS ファイルには、一太郎に付属している「ATOK」という日本語入力システムを、MS-DOS システムに組み込むための指示が登録されており、これによって、Multiplan 上で ATOK を使うことが可能になります。

図 A.1 に見えるのは日本語入力システムに関するものと、MS-DOS システムの動作環境を設定するものだけですが、さきにあげた RAM ディスク以下の各種のプログラムの場合も、同様な形で CONFIG.SYS ファイルに登録します。それらのプログラムは、MS-DOS の起動時に、CONFIG.SYS ファイルの内容の指示に従って MS-DOS システムに組み込まれ、あたかも MS-DOS システムの機能であるかのように動作します。これらのプログラムの実体は、通常の外部コマンドや外部プログラムと同様の、実行可能なマシン語のプログラムですが、そのプログラムの動作状況はまったく異なり、「MS-DOS システムの機能の一部」として動作します。

CONFIG.SYS ファイルによって MS-DOS システムに組み込まれる、このような形式のプログラムのことを、**デバイスドライバ**と呼びます。ここで、「デバイスドライバ」「CONFIG.SYS ファイル」「MS-DOS」、それに「アプリケーションソフト(Multiplan)」との関係を、2 章の図 2.1 を利用して示しましょう(図 **A.2**)。

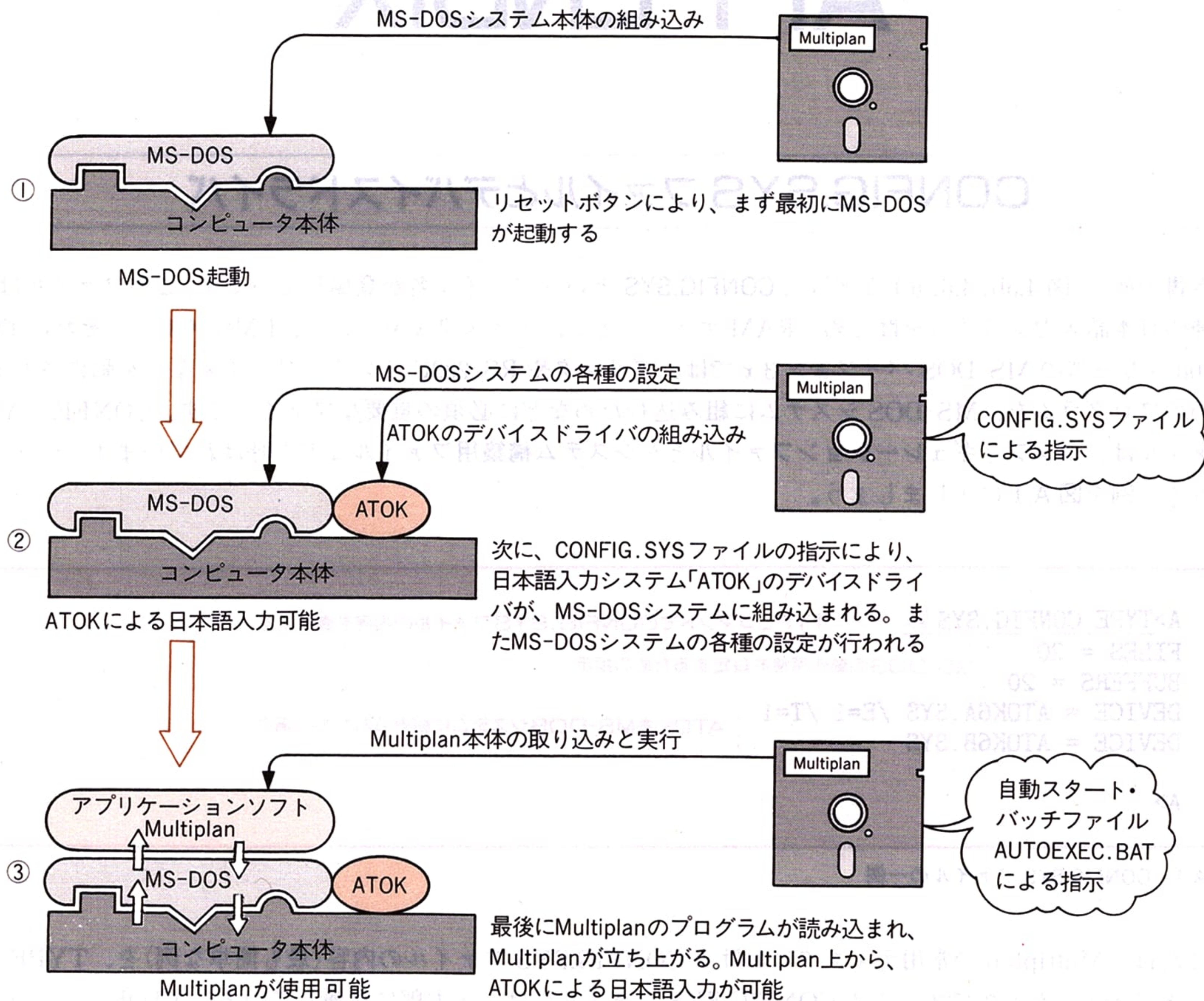



図 A.2　CONFIG.SYS ファイルの働き

この図は、MS-DOS システムが組み込まれている、Multiplan の常用ディスクを起動する過程での CONFIG.SYS ファイルの働きを示しています。1 章の図 1.2、1.3 に示されている Multiplan の起動における内部過程と考えてください。

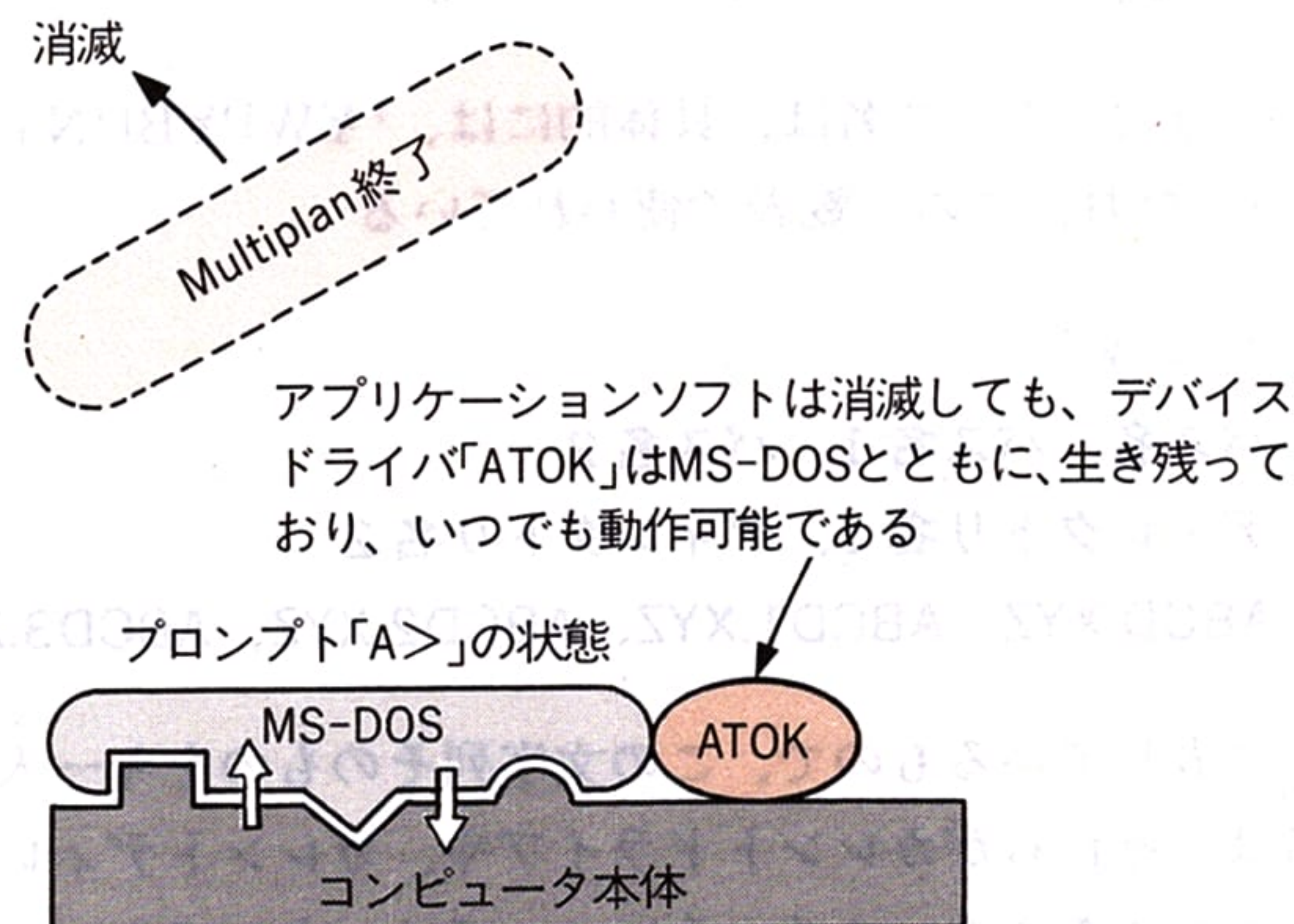
このディスクの CONFIG.SYS ファイルの内容は、さきに TYPE コマンドで表示したものと同じとしましょう。つまり、日本語入力システム「ATOK」のデバイスドライバを組み込む指示と、MS-DOS システムの動作環境を設定する指示とが登録されています(動作環境の設定について詳しくは本書の続編で解説する)。

図の②のように、MS-DOS の本体が起動した直後、CONFIG.SYS ファイルの内容が読み出され、その内容に指示されているデバイスドライバが MS-DOS システムに組み込まれ、同時に MS-DOS システムの環境

の設定が行われます。本書でしばしばお目にかかるMS-DOSのオープニングメッセージと、COMMANDバージョンのメッセージとの間に表示されているメッセージは、この各種のデバイスドライバが組み込まれた際のメッセージなのです。

たとえば、1章の図1.2には、ATOKが組み込まれたことを示すメッセージ、「自由文変換システム ATOK6 ……」が表示されています。また、2章の図2.4には、プリンタと、RS-232Cインターフェイス、NECの日本語入力システムの各デバイスドライバが組み込まれたことを示すメッセージが表示されていますので、再度確認してみてください。

さて、図の③の状態は、アプリケーションソフトであるMultiplanが動作中ですが、このMultiplanを終了するとMS-DOSの状態はどうなるのでしょう。

2章の図2.7に示されているような、Multiplanを終了して、MS-DOSのプロンプト「A＞」が表示されている状態は、実は③の状態から、Multiplanのみを取り去った状態――つまり、Multiplanを立ち上げる前の②の状態とまったく同じなのです(図A.3)。

図A.3　アプリケーションソフトを終了しても生きているデバイスドライバ

ただしここで重要なのは、組み込まれた日本語入力システム「ATOK」は、MS-DOSとともに生き残っていることです。必要であればいつでも日本語入力が可能であり、またMultiplanのあとに、たとえばパソコン通信ソフト「まいとーく」を立ち上げても、「まいとーく」上から日本語入力を行うことができるのです。

『MS-DOSとともに存在する』――これが日本語入力に限らず、すべてのデバイスドライバの特徴です。デバイスドライバは、アプリケーションソフトの起動や終了に関係せず、MS-DOSとともに(あたかもMS-DOSシステムのように)存在し、機能するプログラムです。そして、そのデバイスドライバをMS-DOSに組み込むための指示が、CONFIG.SYSファイルの中の、「DEVICE = ××××」なのです。

主要コマンド一覧表

■ 一覧表の見方

表が煩雑になるのを避けるために、階層ディレクトリに直接関係しない各コマンドの書式は、ルートディレクトリ、あるいはカレントディレクトリ上で実行する場合について示しています。それらが階層ディレクトリに関係する場合は、ここで示されている書式に、パス名を付加する必要があります。

パス名は、ドライブ名とファイル名の間に付加します。その一例を次に示しましょう。1)がここでの書式で、2)はそれにパス名を付加したものです。

1) COPY　*x*：ABCD.XYZ　*y*：
　(ドライブ*x*：上のファイル ABCD.XYZ を、ドライブ*y*：上にコピーする)
2) COPY　*x*：**パス名 1¥ABCD.XYZ**　*y*：**パス名 2**
　(ドライブ*x*：の「パス名 1」で指定されるディレクトリ上のファイル ABCD.XYZ を、ドライブ*y*：の「パス名 2」で指定されるディレクトリ上にコピーする)

また、「パス名 1」や「パス名 2」のパス名は、具体的には、「¥WP¥BUN」のように記述します。詳しくは本文 11 章をご覧ください。なお、この一覧表で使われている

ドライブ名…………*x*：、*y*：
パス名………………**パス名、パス名 1、パス名 2**
ディレクトリ名……**ディレクトリ名 1、ディレクトリ名 2**
ファイル名…………ABCD.XYZ、ABCD1.XYZ、ABCD2.XYZ、ABCD3.XYZ

などは、それぞれを象徴的に表しているもので、この文字列そのものをキー入力するものではありません。また、ドライブ名やパス名は、それらがカレントドライブや、カレントディレクトリの場合は、コマンドライン上の記述を省略することができます。スイッチについては、すべてのコマンドにおいて、たとえば DIR コマンドの「/W」「/P」を、一度に「/W/P」のように、重ねて記述することができます。なお、各コマンドの見出しの右下に「〈3.*x*〉」マークのあるコマンドは、MS-DOS のバージョン 3.*x* 以降に提供されたものですので、バージョン 2.*x* では実行できません。

■ディスク関係コマンド

ディスクのフォーマットコマンド　**FORMAT**　－外部コマンド－

● FORMAT　*x*：
ドライブ*x*：上のフロッピーディスクをフォーマット処理（初期化）する
・スイッチ（コマンドラインの最後に付ける　例…FORMAT　*x*：/S）
/S……フォーマット処理後に、MS-DOS システム（COMMAND.COM を含む）をコピーする
/V……フォーマット処理後に、ボリュームラベルを付ける

ディスクまるごとコピーコマンド **DISKCOPY** －外部コマンド－

●DISKCOPY　*x*：　*y*：

ドライブ*x*：のディスク内容を、ドライブ*y*：上のディスクにまるごとコピーする

ディスクの内容／状態のチェックコマンド **CHKDSK** －ワイルドカード使用可、外部コマンド－

●CHKDSK　*x*：

ドライブ*x*：上のディスクの使用状態（容量等の）をチェックする

●CHKDSK　*x*：ABCD.XYZ

ドライブ*x*：上のディスクの使用状態と、ファイルABCD.XYZの状態をチェックする

・スイッチ（コマンドラインの最後に付ける　例…CHKDSK　*x*：/V）

/V……階層ディレクトリ全体の構造を表示する

/F……修復可能なディレクトリのエラーがあった場合は、自動的に修復する

MS-DOSシステム部のコピーコマンド **SYS** －外部コマンド－

●SYS　*x*：

カレントドライブ上のシステム部（IO.SYS、MSDOS.SYS）を、ドライブ*x*：上のディスクにコピーする（COMMAND.COMはコピーされない）

ディスクのボリュームラベル表示コマンド **VOL** －内蔵コマンド－

●VOL　*x*：

ドライブ*x*：上のディスクのディスク名（ボリュームラベル）を表示する

ボリュームラベルの作成／変更／削除 **LABEL** －内蔵コマンド－

●LABEL　*x*：　〈3.*x*〉

ドライブ*x*：上のディスクのディスク名（ボリュームラベル）を設定／変更／削除する（ボリュームラベルの長さは、半角英数カナ文字ならば11文字、全角文字ならば5文字まで）。実行はメニュー画面による対話形式で行われる

■ファイル関係コマンド

ファイル情報表示コマンド	**DIR**	―ワイルドカード使用可、内蔵コマンド―

●DIR　x：
ドライブx：上のファイル情報を表示する
●DIR　x：ABCD.XYZ
ドライブx：上のファイルABCD.XYZに関する情報を表示する
●DIR　x：パス名￥ディレクトリ名
ドライブx：上の「パス名￥ディレクトリ名」で指定されるディレクトリ上のすべてのファイルの情報を表示する
・スイッチ（コマンドラインの最後に付ける　例…DIR　x：/W）
/W……ファイル名だけのワイド表示
/P ……1画面ごとに表示

ファイルのコピー（転送）コマンド	**COPY**	―ワイルドカード使用可、内蔵コマンド―

●COPY　x：ABCD.XYZ　y：
ドライブx：上のファイルABCD.XYZをドライブy：上にコピーする
●COPY　x：ABCD1.XYZ　y：ABCD2.XYZ
ドライブx：上のファイルABCD1.XYZを、ドライブy：上にABCD2.XYZのファイル名でコピーする
●COPY　x：パス名￥ディレクトリ名　y：
ドライブx：上の「パス名￥ディレクトリ名」で指定されるディレクトリ上のすべてのファイルを、ドライブy：上にコピーする
●COPY　x：パス名1￥ディレクトリ名1　y：パス名2￥ディレクトリ名2
ドライブx：上の「パス名1￥ディレクトリ名1」で指定されるディレクトリ上のすべてのファイルを、ドライブy：上の「パス名2￥ディレクトリ名2」で指定されるディレクトリ上にコピーする
●COPY　x：ABCD.XYZ　PRN
ドライブx：上のファイルABCD.XYZを、プリンタに出力する
●COPY　CON　x：ABCD.XYZ
キーボードからの入力を、ドライブx：上にファイルABCD.XYZを作成して格納する
・スイッチ（コマンドラインの最後に付ける　例…COPY　x：ABCD.XYZ　y：/V）
/V……書き込み直後に読み出して原データと比較チェックを行う（ただし、ディスク間コピーの場合のみ）

ファイル削除コマンド	**DEL**	―ワイルドカード使用可、内蔵コマンド―

●DEL　x：ABCD.XYZ
ドライブx：上のファイルABCD.XYZを削除する
●DEL　パス名
ドライブx：上の「パス名」で指定されるディレクトリ上のすべてのファイルを削除する

ファイル名変更コマンド **REN** －ワイルドカード使用可、内蔵コマンド－

●REN　x：ABCD1.XYZ　ABCD2.XYZ

ドライブx：上のファイルABCD1.XYZのファイル名を、ABCD2.XYZに変更する

文字ファイル画面表示コマンド **TYPE** －ワイルドカード使用不可、内蔵コマンド－

●TYPE　x：ABCD.XYZ

ドライブx：上の文字ファイルABCD.XYZの内容を画面に表示する

2つのファイルの内容比較コマンド **FC** －ワイルドカード使用不可、内蔵コマンド－

●FC　x：ABCD1.XYZ　y：ABCD2.XYZ

ドライブx：上のファイルABCD1.XYZと、ドライブy：上のファイルABCD2.XYZを比較して、その差異を表示する

・**スイッチ**（FCの直後に付ける　例…FC/B　x：ABCD1.COM　y：ABCD2.COM）

/B……バイナリファイル（.COMや.EXEファイルなど）を対象にする場合（バイト単位で表示する）

文字列サーチコマンド **FIND** －外部コマンド－

●FIND　"ABCDEF"　x：ABCD1.XYZ　y：ABCD2.XYZ　z：ABCD3.XYZ

ドライブx：、y：、z：上の文字ファイルABCD1.XYZ、ABCD2.XYZ、ABCD3.XYZから、文字列「ABCDEF」をサーチし、存在する行をすべて表示する（フィルタとしての使い方は、「フィルタ関係コマンド」の項を参照）

・**スイッチ**（FINDの直後に付ける　例…FIND/V　"ABCDEF"　x：ABCD1.XYZ～）

/V……サーチする文字列を含まない行を表示する

/C……含まれる行の数を表示する

ファイルの属性の変更／表示 **ATTRIB** －ワイルドカード使用可、外部コマンド－

〈3.x〉

●ATTRIB　x：ABCD.XYZ

ファイルABCD.XYZの属性を表示する

●ATTRIB　＋R　x：ABCD.XYZ

ファイルABCD.XYZの属性をリードオンリー（読み出しのみ可能）にする

●ATTRIB　－R　x：ABCD.XYZ

ファイルABCD.XYZのリードオンリー属性を解除する

■階層ディレクトリ関係コマンド

カレントディレクトリ変更コマンド	**CD**	―内蔵コマンド―

●CD　x：パス名￥ディレクトリ名
ドライブx：上のカレントディレクトリを「パス名￥ディレクトリ名」で指定されるディレクトリに移す

ディレクトリ作成コマンド	**MD**	―内蔵コマンド―

●MD　x：パス名￥ディレクトリ名
ドライブx：上に「パス名￥ディレクトリ名」で指定される、新しいディレクトリを作成する

ディレクトリ削除コマンド	**RD**	―内蔵コマンド―

●RD　x：パス名￥ディレクトリ名
ドライブx：上の「パス名￥ディレクトリ名」で指定されるディレクトリを削除する。ただし、カレントディレイトリと、ファイルやサブディレクトリが存在しているディレクトリは削除できないので注意

外部コマンドを実行する時の サーチ対象ディレクトリ指定コマンド	**PATH**	―内蔵コマンド―

●PATH　x：パス名１￥ディレクトリ名１；y：パス名２￥ディレクトリ名２；～
ドライブx：、y：～上の「パス名1」「パス名2」～で指定される「ディレクトリ名1」、「ディレクトリ名2」内の外部コマンド／プログラムについて、ファイル名のみのキー入力による実行を可能にする（設定しない場合はカレントディレクトリ上のコマンド／プログラムのみ実行可能）

●PATH
現在設定されているサーチパスを表示する

●PATH；
現在設定されているパスをすべて解除する

ディレクトリ構造の表示	**TREE**	―外部コマンド―

●TREE　x：
〈3.x〉
ドライブx：上のすべてのディレクトリのパス名を表示する

・スイッチ（コマンドの最後に付ける　例……TREE　x：/F）
/F……各ディレクトリのファイル名も表示する

■フィルタ関係コマンド

文字列サーチコマンド	**FIND**	－外部コマンド－

●FIND　"abcdef"

フィルタとして使用する場合、入力データから文字列「abcdef」をサーチして、存在するすべての行を出力する。通常のコマンドとして使用する場合は「ファイル関係コマンド」の項を参照

・**スイッチ**（FINDの直後に付ける　例…FIND/V　"abcdef"）

/V……サーチする文字列を含まない行を表示する

/C……含まれる行の数を表示する

1画面単位の表示コマンド	**MORE**	－外部コマンド－

●MORE

フィルタとして使用する。画面のスクロールを1画面ごとに停止する

データのソートコマンド	**SORT**	－外部コマンド－

●SORT

フィルタとして使用する。入力データの各行の1文字目から各行をソートする

・**スイッチ**（SORTの直後に付ける　例…SORT/R）

/R　……逆順ソートする

/＋n……入力データの各行のn文字目から各行をソートする

■その他のコマンド

日／時の表示／設定コマンド	**DATE、TIME**	－内蔵コマンド－

●DATE

日付を表示し、必要があればその変更をする（変更する場合は表示と同じ形式で新しい日付を入力する。しない場合はそのまま⏎を入力する）

●TIME

時刻を表示し、必要があればその変更をする（変更する場合は表示と同じ形式で新しい時刻を入力する。しない場合はそのまま⏎を入力する）

画面のオールクリアコマンド	**CLS**	－内蔵コマンド－

●CLS

画面の表示をオールクリアする

あとがき

本書をもとに、MS-DOSに取り組んでこられたみなさんは、ここでの知識を、さまざまな仕事に活用していることと思います。MS-DOSによってOSというものの存在に気づき、コンピュータやソフトウェアの本質に目覚めた人の多くは、『こんなことならもっと早く学んでおけばよかった』という気持ちを持たれたのではないでしょうか。それはきっと、MS-DOSを理解することの大きな意味を、実際に体験できたことによるでしょう。

本書で得られたOSの知識は、今後、アイコンなどを使ったグラフィック的なマン-マシン・インターフェイス(コンピュータの操作法)がどれほど発達しようと、コンピュータの「活用者」には必ず必要なものです。そしてMS-DOSは、今後とも長期にわたって、私たちのパーソナル・コンピュータ全般の重要な柱として存在します。

さて、「まえがき」でも述べましたが、今、MS-DOSの世界に踏み込んだみなさんは、これからのコンピュータ社会を歩くための最初のキップを手にしたのです。

索　引

MS-DOS、Microsoft Multiplan、Microsoft C Compiler、Microsoft Quick C Compiler、Microsoft Quick BASIC Compilerは、米国Microsoft社の商標です。
IBM、IBM PC、PC-DOSは、International Buisnedd Machines社の登録商標です。
CP／M、CP／M-86は、米国Digital Reserch社の登録商標です。
UNIXオペレーティングシステムは、AT&Tのベル研究所が開発し、AT&Tがライセンスしています。
Visi Calcは、米国Visi Corp社の登録商標です。
Turbo Cは、米国Borland International社の登録商標です。
一太郎、ATOK6、ATOK7は、㈱ジャストシステムの登録商標です。
新松、桐は、㈱管理工学研究所の登録商標です。
Lotus 1-2-3は、米国Lotus Development社の商標です。
informix-SQLは、米国Informix Software社の商標です。
Mifes-98は、メガソフト㈱の製品です。

入門MS-DOS　改訂新版
アスキー・ラーニングシステム①入門コース

1984年7月7日　初版発行
1989年12月21日　第3版第1刷発行
1990年4月21日　第3版第3刷発行
定価1,600円（本体1,553円）

著　者　村瀬康治（むらせやすはる）
発行者　塚本慶一郎
発行所　株式会社アスキー
〒107-24　東京都港区南青山6-11-1スリーエフ南青山ビル
振　替　東京4-161144
TEL (03)486-7111（大代表）
情報TEL (03)498-0299（ダイヤルイン）
出版営業部TEL (03)486-1977（ダイヤルイン）

制　作　株式会社 GARO
印　刷　株式会社 加藤文明社印刷所

編　集　　小栗葉子

ISBN 4-7561-0016-3　　Printed in Japan

● 11072／11380／11490